Call-A.P.P.L.E.™

Volume IV, Number 3 March 1982 $2.50 ($3.00 Canada, £1.75 UK.)

A IS FOR APPLE:
Pop Art
Page 9

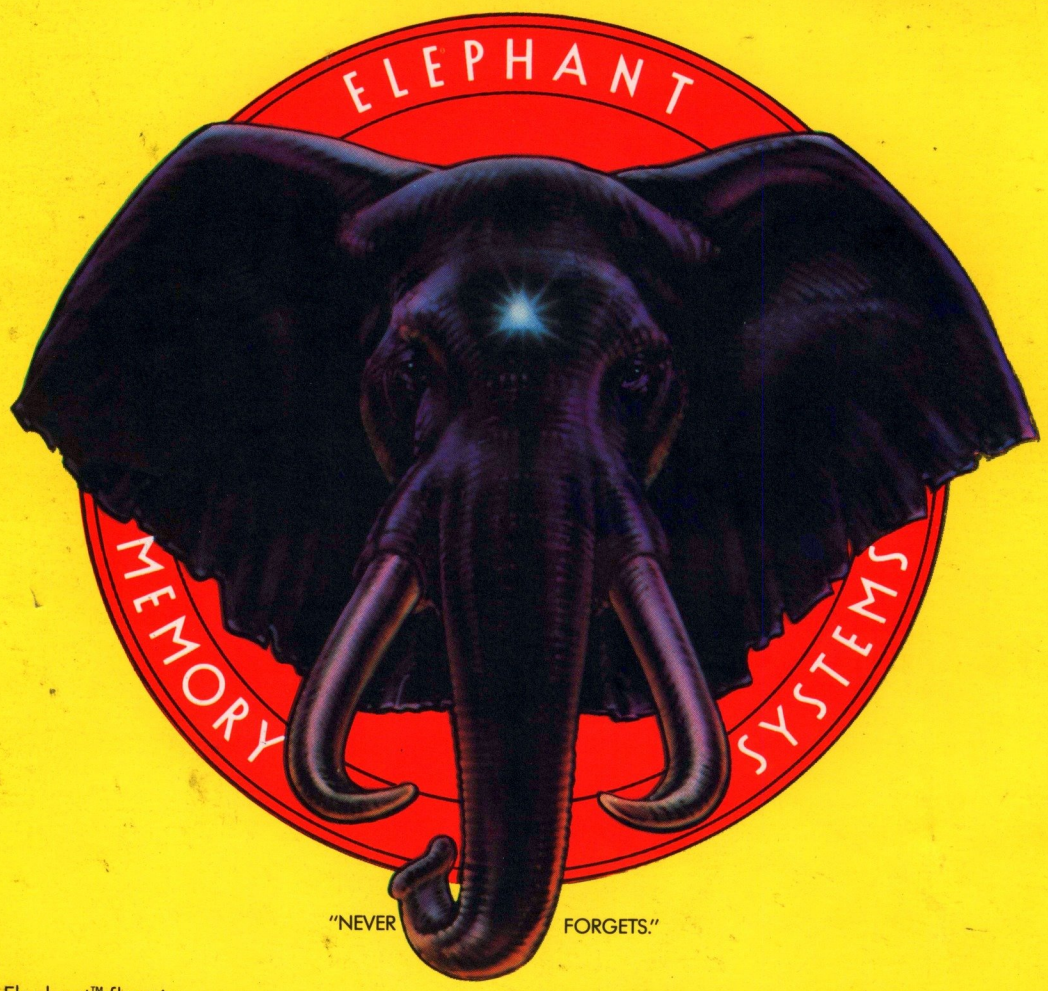
EXPLORING THE WORLD OF APPLE:
Get The Most From Your Diskettes
Page 13

FROM THE VERY CORE OF APPLE:
Applesoft's CHAR-GET Routine
Page 21

SPECIAL MATH SECTION:
Getting the High Sine
Page 27

Apple PugetSound Program Library Exchange

REMEMBER.



"NEVER FORGETS."

Elephant™ floppies.
They're guaranteed to meet or beat every industry standard for quality. They come standard with reinforced hub rings at no extra cost. They come in every popular 5¼" model, in both hard and soft sector. And they sell at some of the lowest prices in the business.
Elephant Flexible Disks.
They're heavy duty. They work for peanuts.
They never forget. Get yourself a trunkful.

HEAVY DUTY DISKS.

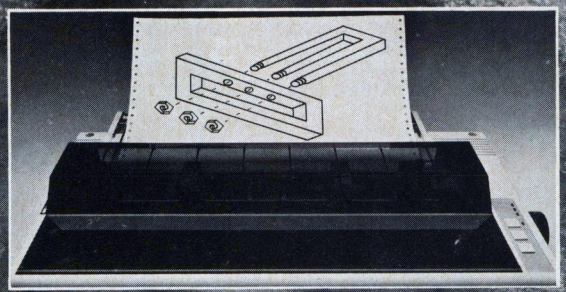
Distributed Exclusively by **Leading Edge Products, Inc.**, 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.

IMAGINE IT..

CAPTURE IT.

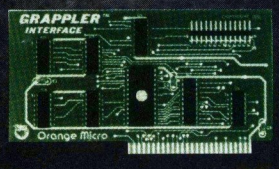
The Grappler™ is the most intelligent Apple interface available. Simple commands from the keyboard or user program, are all you need to dump screen graphics to your printer. The Grappler's exclusive EPROM chip makes high resolution graphics that easy. There are Grappler versions to accommodate the Anadex, Epson Series,* IDS Paper Tigers, Centronics 739, NEC 8023, C. Itoh Pro-writer, Okidata 82A,* Malibu 200, and future graphic printers.

*Requires Graphics Upgrade.



ACTUAL APPLE II PRINTOUT USING GRAPPLER™ AND EPSON MX100

With The Grappler™ Printer Interface



The Grappler's features include:

- Graphics Screen Dump
- Inverse Graphics
- Emphasized Graphics
- Double Size Picture
- 90° Rotation
- Center Graphics
- Chart Recorder
- Mode Block Graphics
- Bell Control
- Skip-over-Perf
- Left and Right Margins
- Variable Line Length
- Text Screen Dump.

The Grappler™ works with Pascal and CPM.

Orange Micro inc.

3150 E. La Palma, Suite G, Anaheim, CA 92806

(800) 854 8275 TOLL FREE

(714) 630 3322 CA; AK; HI.

© 1982, Orange Micro, Inc.

APPLE PUGETSOUND PROGRAM LIBRARY EXCHANGE:

CLUB INFORMATION

GENERAL OFFICES
304 Main Ave. S., Suite 300
Renton, WA 98055 — (206) 271-4514
Hours: 8-4 Monday Through Friday
Member Services/Dealer Orders

EDITORIAL DEPARTMENT
6708 39th Avenue S.W.
Seattle, WA 98136 — (206) 932-6588
Hours: 9 AM - 3 PM, 6 PM - 10 PM, daily

ADVERTISING REPRESENTATIVES
Grawin Publications
1020 Lloyd Building
Seattle, WA 98101 (206) 223-0861

MEETING NOTICES

7:30 PM, Tuesday, March 16, 1982
A.P.P.L.E. Office
304 Main Ave., Suite 300, Renton
Guest Speaker
Jim Egan
Business Forms Considerations

7:30 PM, Tuesday, April 20, 1982
Omega Computers
1032 N.E. 65th Street, Seattle
Guest Speaker
Jim Grimshaw
Memorex Diskettes

7:30 PM, May 18, 1982
Computer Empire
616 S.W. 152nd St., Seattle

BULLETIN BOARDS

A.P.P.L.E. CRATE
(206) 935-9119
(4 pm - 9 am Pacific Time)
Apple CRATE][— Empire
(206) 244-5438
(24 hours)

CALL —A.P.P.L.E. HOT LINE

(206) 932-6588
Hours: 9 AM - 3 PM and
6 PM - 10 PM 7 days per week
Pacific Time

See page 91 of the Nov/Dec '81 issue for information on use of the Hotline.

Membership Information

Please direct telephone questions on membership and orders to (206) 271-4514.

One Time Apple—Cation Fee	$25.00
Annual Dues	$20.00
Total	$45.00

Additional postage required for foreign memberships.

Membership is on an annual, 12 month basis, commencing with the month of enrollment. Upon receipt of total membership fees, new members will receive an Apple—Cation blank, order form and description sheet of club products.

Please allow up to 30 days for a turn around on membership and orders.

Call —A.P.P.L.E. is mailed free twelve times per year as benefit of membership. Subscriptions are not available.

1982 back issues available at $2.50 each.

Call —A.P.P.L.E. bound volumes for previous years are available as indicated on the member order form on the last page.

Prospective authors please send self-addressed stamped envelope (37¢ postage) to Editorial Dept. for a copy of Author's guidelines.

Cover and interior design by Vic Warren Design.

Officers and Directors

Dick Hubert, President	(206) 271-6939
Bob Huelsdonk, Vice President	(206) 362-4910
Dave Lingwood, Secretary	(206) 244-9360
Bob Clardy, Treasurer	(206) 641-1917
Val J. Golding, Director	(206) 932-6588
Ken Smith, Director	(206) 322-4199
Fred Merchant, Director	(415) 921-7792

Office Staff

Dick Hubert	Executive Director
Kazi Ahmed	Administrative Assistant
Sally Davies	Admin. of Information Systems
Arliss Sword	Dealer Orders
Patti Hoffmann	Shipping, Receiving
Gunter Hirt	Bookkeeping, Member Records
Joff Morgan	Technical Support
Linda Wallin	Special Projects Coordinator
Kris Mortenson	Order Processing
Gayle Rank	Secretary/Receptionist
Mike Christensen	Systems Development
Gina Mocko	Shipping/Receiving

Magazine Staff

Val J. Golding	Editor	(206) 932-6588
Kathryn Hallgrimson	Asst. to the Editor	(206) 932-8288
Ron DeGroat	Pascal Editor	(303) 442-4794
Kevin O. Donohoe	Production Editor	(206) 223-0861
Paul Mosher, M.D.	Staff Writer	(518) 765-4118
Lee Reynolds	Staff Writer	(305) 722-2639
Charles Rusch	Staff Writer	(503) 484-4618
Ralph H. Swerdlow, M.D.	Staff Writer	(916) 920-8709
Grawin Publications	Production, Advertising	(206) 223-0861
David B. Garson	Program Editor	(714) 452-7524

Consultants

Bob Huelsdonk	Apple ///	(206) 362-4910
Don Williams	Assembly	(206) 242-6807
Michael Thyng	Business	(206) 524-2744
Ted Perry	C.A.I.	(916) 485-1690
Jeff Finn	Data Communications	(206) 325-0430
Murray Arnow	Engineering Applications	(312) 291-9210
Cornelis Bongers	European Consultant	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Randy Halvorson	Hardware Interfacing	
Ed So	Hi-Res	(408) 267-3448
Alan Winston	Languages	(206) 789-1512
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Tom Woteki	Pascal (Eastern U.S.)	(202) 547-0984
Carl Winter	Pascal (Western U.S.)	(206) 454-2720
Hersch Pilloff	Stock Market	(301) 292-3100
Mark Robbins	Telecommunications	(303) 755-6440
Clifton Howard, M.D.	Utilities	(201) 768-9274
Jim Hockenhull	Video Interfacing	(509) 334-4137
Dick Curtiss	Z-80 Card, CP/M	(206) 784-8018

Call-A.P.P.L.E.™

Volume 5, Number 3 March 1982/Menu Val J. Golding, Editor

ENTRÉES

9 A IS FOR APPLE:
Pop Art
Ralph Swerdlow, M.D./Call —A.P.P.L.E. Staff Writer

13 EXPLORING THE WORLD OF APPLE:
Get the Most From a Diskette
Clay Ruth

21 FROM THE VERY CORE OF APPLE:
Applesoft's CHARGET Routine
Cornelis Bongers/Call A.P.P.L.E. Staff Writer

27 SPECIAL MATH SECTION:
Getting the High Sine: Parsing String Formulas
Charles Kluepfel

31 Trig or Treat
Raymond Usher

33 APPLEMASH:
Mathemagics
Michael Thyng

37 Mod Art
Wayne Eastwood

À LA CARTE

41 A CALL —A.P.P.L.E UTILITY:
MX 100 Hi-Res Dump
Bruce C. Detterich

51 CALL—A.P.P.L.E. CONTEST WINNERS:
Out of Sorts

53 HEXADECACOLOR DOUBLE BUBBLE DEMO SORT
Scott Simpson/1st prize winner

57 Call —APPLE CONTEST ENTRY COMMENTS
Luis A. Arana

65 NOTES FROM A NOVICE:
Charles W. Rusch/Call —A.P.P.L.E. Staff Writer

70 A CALL —A.P.P.L.E. UTILITY:
LCASE Killer
Eric Lambreht

HORS d'OEUVRES

2 A.P.P.L.E. INFORMATION

5 The Editor Bytes Back
Val J. Golding

7 Bytes From A.P.P.L.E.
The Editor

73 A.P.P.L.E. Meeting Minutes
Dave Lingwood/Secretary

74 WRITE A.P.P.L.E.
The Readers

76 Vandal Foilers Foiled, Editor Omits Program

79 Advertiser's Index

79 A.P.P.L.E. Order Form

80 Call —A.P.P.L.E. Unclassifieds

Apple user groups may reprint without prior permission any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commercial applications of programs herein, the author should be contacted c/o General Office.

Entire contents copyright© 1982 by

Apple PugetSound Program Library Exchange

NEW...FOR APPLE II[1]

FROM PROMETHEUS

VERSAcard

1. Serial Input/Output Interface
2. Parallel Output Interface
3. Precision Clock/Calendar
4. BSR Control

NEW 4 in 1 CARD

ALL ON ONE CARD
with true simultaneous operation . . .at a down-to-earth price!

CHECK THESE FEATURES: True simultaneous operation! VERSAcard's unique hardware selection circuitry make your Apple "think" that separate cards are installed. This allows VERSAcard interface ports to be compatible with existing software such as APPLE PASCAL, Microsoft Softcard[2], and most other Apple software.

The Serial Input/Output port is RS-232C standard with crystal controlled baud rates from 19.2K to 110 baud. You can connect with a CRT terminal, line printer, or virtually any RS-232 modem. VERSAcard's "Terminal" mode, along with modem, connects you with the *Source* or a variety of other computers. "Terminal" mode also allows you to output to both printer and display simultaneously.

The versatile **Parallel Output Interface** is configured for the Centronics standard and is easily configured for other standards. Simply hookup your **Epson, Okidata, IDS,** or other printer and begin printing.

Precision Clock/Calendar. With all standard real-time clock/calendar functions and one second resolution. Battery back-up supplied. And, best of all, it's **software compatible with Thunderclock.** This makes numerous other software packages immediately usable. Interrupts are supported.

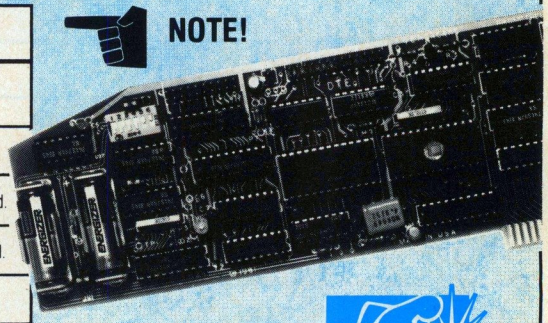
BSR Control. Add an ultrasonic transducer, and VERSAcard provides you with remote control through your BSR X-10 Home Control System.

Onboard firmware provides optional auto line feed, video, paging, terminal mode, lower to upper case, 8th bit set or clear ... and more. These versatile device drivers permit connection to virtually any standard peripheral. The Prometheus engineering staff is available to help. With any peripheral and special device interface problems. Call us, we'd like to hear from you.

VCD-1 VERSAcard @ $249.00 each.

Compare Key Features	VERSAcard	CPS MultiFunction Card
Parallel, Serial and Realtime Clock Interfaces?	YES	YES
BSR Control Interface?	YES	NO
Compatible w/ Apple Pascal?	YES	NO. Special Pascal disk required.
Compatible with CP/M?	YES	NO. Special CP/M disk required.
Interrupts?	YES	NO
Apple Modification?	NO	YES
"**NEW** . . . Applications/Diagnostic disk—available 1st quarter '82"		

NOTE!



Dealer inquiries invited.

VERSAcard—the only Apple peripheral card you're likely ever to need!

All Prometheus boards are completely tested and burned-in prior to shipment. A one-year warranty covers parts and labor.

Registered Trade Marks: (1) Apple Computers, Inc. (2) Microsoft Consumer Products

PROMETHEUS

PROMETHEUS PRODUCTS INCORPORATED

45277 Fremont Blvd. • Fremont, CA 94538 • (415) 490-2370

Ask about the new PROMETHEUS 128K RAM card.

THE EDITOR BYTES BACK

by the editor

CALL -A.P.P.L.E. staff writer **Ralph Swerdlow** again appears at the top of our story list and our cover with another beginner-oriented feature as **A IS FOR APPLE** brings to light the underused POP command in **Pop Art.**

How would you like to be able to use up 70 additional sectors on a DOS 3.3 diskette? **EXPLORING THE WORLD OF APPLE** in this issue tells you how, with detailed instructions in newcomer **Clay Ruth's Get the Most**.

The **Applesoft CHARGET Routine** is thoroughly examined, and some improvements suggested as staff writer **Cornelis Bongers** once more takes the reins of **FROM THE VERY CORE OF APPLE.** If you are into Applesoft internals, then don't miss this one.

A last minute addition to this edition was the inclusion of a **SPECIAL MATH SECTION.** It comprises three semi-related features: **Charlie Kluepfel's Getting the High Sine,** an ampersand utility to parse a string formula into Applesoft, **Trig or Treat, Ray Usher's** high level math algorithms, and **APPLE-MASH: Mathemagics; Mike Thyng** describes how to obtain 18 digit (and more) precision.

Out of Sorts is a detailed explanation of what went on in our September contest program by **Tom Crawford,** written by first prize winner **Scott Simpson,** with additional comments by **Luis Arana,** a member from Argentina.

Bruce Detterich returns with a postscript to his February review of the MX100, and offers **Epson Hi-Res Dump.**

As always, we have more for you; just turn back to page three to see. (Take three and see?) What's more, we have a whole pile of outstanding features that we couldn't possibly fit into this issue, but which will appear in April and coming months. For example:

Dan Paymar will be back with **DICE,** a comprehensive keyboard utility which can be used with **AplCase** and other lower case display chips, and we will also be telling you about **A.P.P.L.E.** member number 10,000!

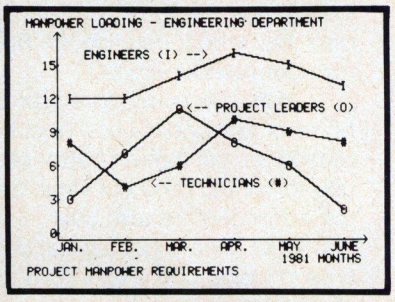
Gosh. We could go on for another page or two and tell you about what is coming up, but a much better idea is for you to tell **us.** Why not drop a note to the editorial office with your "wish list," and let us know what we can do to continue bringing you a well balanced diet of Apples.

Dickens Data Systems,
your graphics specialists, presents: The professionally oriented graphics package

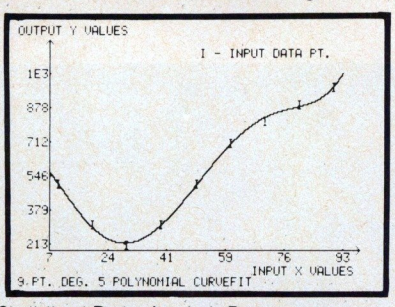
THE SUPERPLOTTER©

version 2.2

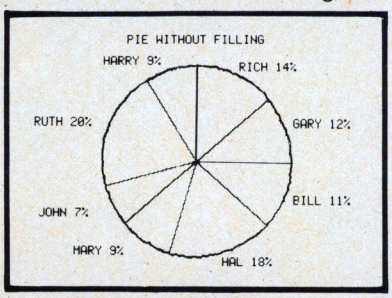
For use with Apple™ Computer Systems



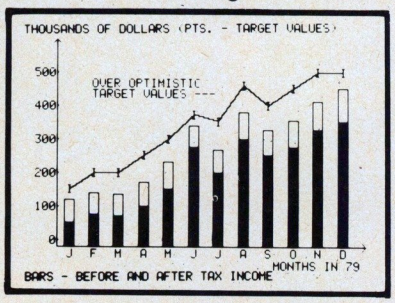
General Purpose Plotting



Statistical Data Analysis Forecasting Tool



Decision Making Tool



Extensive Overlay Capabilities

The Superplotter is a highly versatile business, engineering, educational, math and graphics applications package featuring:

Pie Graphs

Standard Bar Graphs

Point and Line Graphs

Graphics Display of any Mathematical Function

Least Squares Polynomial Curvefit Generation

Keyboard Image Shape Tables and User Tutorial

Automatic Graphics Disk Storage and Recall

Data File Editor

Automatic Disk Storage and Recall of Editor Data Files

Overlay Modes

Graphics Screen Text Editor

Data may be input directly or from previously generated data files. Comprehensive documentation is provided including printer dump instructions and directions for accessing data bases created by other programs. Requires Applesoft and one DOS 3.3 disk drive.

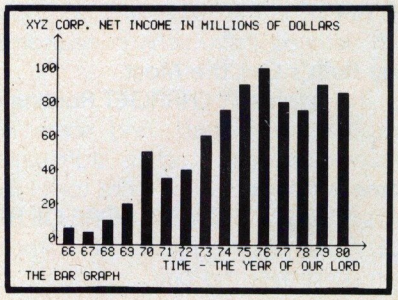
$69.95

See your local dealer for a demonstration or order direct from:

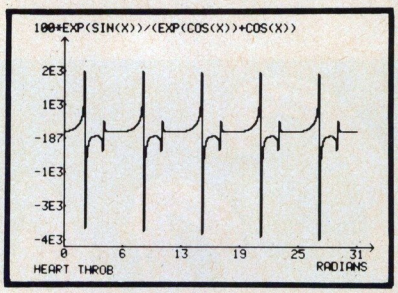
Dickens Data Systems
478 Engle Drive
Tucker, Georgia 30084
(404) 491-7905
outside Georgia: **(800) 241-6753 ext. 503**

C.O.D., VISA, & Mastercard accepted.

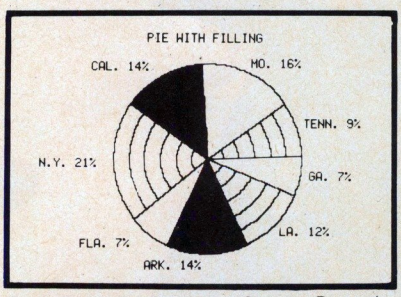
Mail Orders: Add $2.50 for shipping and handling ($5.00 outside the USA)



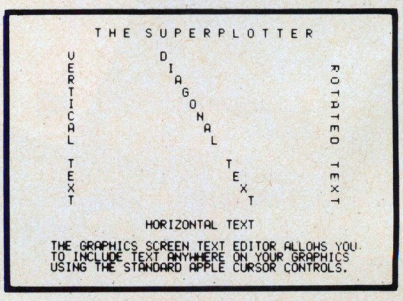
Visual Aid for Presentations



Plot Anything



Perfect for Professional Quality Reports



Plot Keyboard Characters on Any Graphics Display Simply

BYTES FROM A.P.P.L.E.

By Val J. Golding

HARDWARE is our keyword (token? [sorry, we couldn't stop ourselves]) for March. With pleasure, we announce **Ap.Controller** to go along with our **Ap.Drive.** This is a disk controller card which can switch select 13 or 16 sector user-supplied DOS, an excellent way to defeat the time-consuming method of booting on the BASICS disk. Ap.Controller will sell to members for just $95. Moreover, we will make it available as an ongoing "special" in conjunction with the $395 Ap.Drive, with a combination price of just $475.

DICE stands for "Dan's I/O Control Enhancer," which we have added to our Diskpak category for just $12.50. DICE is a complete system for users of **ApLCase** and other lower case display chips which offers easy entry of any ASCII character, upper or lower case, and other goodies. If you get an ApLCase1 (for older Apples) you can add just $7.50 for a copy of DICE. If you have a Rev7 or later Apple and require the ApLCase2, you can get it and DICE for a total of $37.50. Note there is no time limit on either of these specials; they will continue indefinitely.

Big Mac has already achieved a reputation as a real winner; where else can an A.P.P.L.E. member find a first class assembler for $22.50 that does everything Mac does, and now we can tell you about the Language Card version, which will be available around the end of the month at an introductory price of $27.50.

Big Mac.LC adds file linking and nestable macros and many other other refinements to the already powerful facilities of Big Mac, to say nothing of the extra RAM space available or the PLE-type editing features. Our schedule calls for having this super product ready about the end of March. Keep your fingers crossed and your keyboard warm.

Here's a "sneaky" you won't find on the order form. **Apmail][** is our established mailing list/label printing program that handles up to 650 records on one diskette, and will search or sort on up to ten different fields, merge files and other similar niceties. The regular price is $22.50, but if you write or call by 4/30/82 and ask for the special price, it's yours for only 20 bucks. Note this will not appear on the order form or as an ad, it's up to you sharp-eyed members to pick up on this one.

Our closeout on **Program Line Editor** continues this month. While the documentation lasts, you can save $5 on this top rated diskette while we are working on the new version. (Sorry, updates will *not* be available), so quick, grab the order form in the back and take advantage of this offer for $20.

Sometimes we really run into huge unplanned delays with some of our products. This was the case with *Bob Huelsdonk's* **Word Weaver ///,** the Apple /// word processor, but we're happy to say that we are now shipping the program with preliminary documentation which will be updated soon. In the meantime, you can still get in at the introductory price of $60.

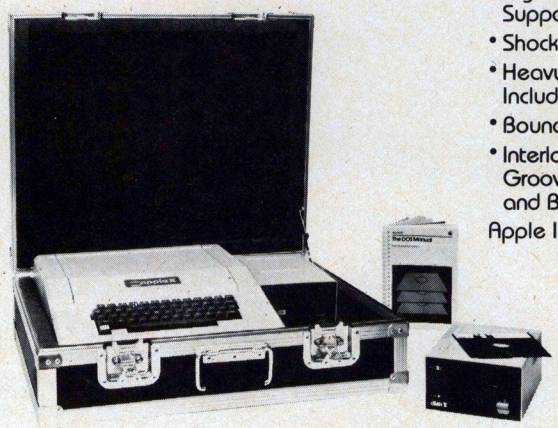
What's Faire is Faire

We will be looking forward to seeing you and meeting you at the West Coast Computer Faire in San Francisco later this month; we plan also to attend several of the Applefests and other upcoming shows. We will also have a booth at many of these shows, which will provide members with the opportunity to say hello to some of our staff whom they have only met on the phone, and at the same time to order and pick up A.P.P.L.E. products on the spot. Non-members may also find this is the ideal time to join, and while plans are not finalized at this writing, we may offer some type of special for those joining at the shows.

FLIGHT FORM

BY CASES, INC.

DESIGNED TO PROTECT YOUR COMPUTER



Features —

- Rigid Shell Made of Plywood Supported High Impact ABS.
- Shock Resistant Foam Lining.
- Heavy Duty Hardware Includes Key Locking Latches
- Bound Metal Edges.
- Interlocking Tongue and Groove Extrusion, Mating Lid and Bottom.

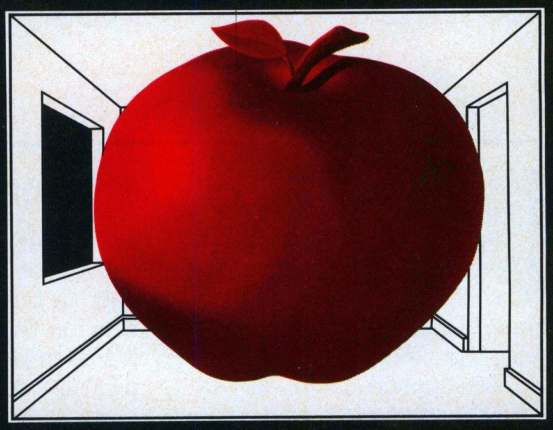
Apple II Plus 2 Drives $175.00

OTHER MODELS AVAILABLE

CLUB AND DEALER DISCOUNTS AVAILABLE

CASES, INC. P.O. Box 33820 Seattle, WA 98133 (206) 365-5210

EXPAND YOUR APPLE



Z-CARD

CP/M FOR THE APPLE II

The Z-CARD is your key to CP/M - the largest software base available with over 2,500 programs for your word processing, forecasting, data management, and accounting applications. And now the Z-CARD is available separately or as part of the total package...The **New** Synergizer.

For the first time ever...a Z-80 interface for the Apple II, integrated with peripheral boards to provide a powerful CP/M operating system. The new ALS SYNERGIZER adds the field-proven SMARTERM (80 column video board) and ADD-RAM (16K RAM expansion) to complete the system. SYNERGIZER software ties it all together - easy for you to learn, easy for you to use.

Don't wait!

Synergize **your** Apple and open the door to CP/M applications today.

Available now through local Distributors and Dealers. For more information, call **800-538-8177.**
In California call **408-730-0306.**
TELEX: 4990514

als Advanced Logic Systems

1195 E. Arques Avenue, Sunnyvale, CA 94086

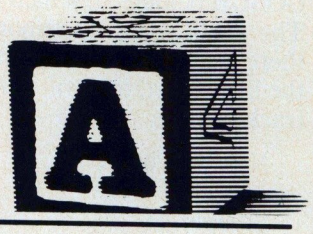
Apple and Apple II ® Apple Computer, Inc., CP/M ® Digital Research, Inc.

© 1982 Advanced Logic Systems

A IS FOR APPLE:

Pop Art

By Ralph H. Swerdlow, M.D./Staff Writer



THERE is a strange, infrequently used command available to the Apple user called "POP". It is a useful command, and should be part of every Apple programmer's knowledge. Perhaps the reason that POP is seldom used is that it is poorly described in the Apple Reference Manual, which says, "A POP has the effect of a RETURN without the branch." What does that mean? In this article I will give you an explanation and an example of what POP does and afterwards, you should be able to utilize this command in your programs.

First, let us look at the GOSUB routine:

```
 5 TEXT:HOME
10 PRINT "10"
20 PRINT "20"
30 PRINT "30"
40 PRINT "40"
50 GOSUB 1000
1000PRINT "50"
```

Line 5, of course, clears and sets the screen to full width. Lines 10,20,30 and 40 print their respective values, and line 50 goes to the subroutine at line 1000 which prints "50." Note that there is no RETURN listed. Run this program, and see that it prints out the values and then ends. The GOSUB in this example, acts exactly like a GOTO.

Now, let's insert a RETURN by adding line 1010:

```
1010 RETURN
```

and also add line 60:

```
60 PRINT "60"
```

When you run this program "10,20,30,40,50 and 60" will be printed as expected, but then "50" will be printed again and will be followed by "?RETURN WITHOUT GOSUB ERROR IN 1010." What happened is that the Apple printed each value in lines 10 through 40, went to the Subroutine at 1000 and printed "50." After the RETURN in line 1010, the computer went back to the very next line after the GOSUB command, line 60, and printed "60." The Apple then continued to line 1000 and printed "50" once more, and then to line 1010 where the command "RETURN" is located. But the RETURN had already been used (no RETURN address remained in memory); thus the error message was printed!!

Here is how to prevent this from happening. Add line 70:

```
70 PRINT "70":END
```

This tells the computer to END after printing "70." The program will *not* continue to lines 1000 and 1010. RUN with these new lines and verify for yourself that it is true. The lesson to be learned is that a program must have some sort of command in it to prevent going to the GOSUB Routine a second time! If it does not, the error message will be printed.

Now change line 1010 to:

```
1010GOSUB 2000
```

and add these lines:

```
1020PRINT "100"
1030RETURN
2000PRINT "80"
2010PRINT "90"
2020RETURN
```

Line 1010 directs the program to another subroutine located at line 2000, which prints "80." Line 2010 prints "90" and line 2020 is the command to RETURN to the very next line following the last GOSUB command executed,in this case line 1020. Here "100" is printed and the program continues to line 1030 which directs the program back to line 60, the line after the first GOSUB instruction! If this program is now run, the numbers 10 through 100 will be printed in this sequence: 10,20,30, 40,50,80,90,100,60 and 70 and then will end.

You can use the command "TRACE" to follow each step in the program. The command "SPEED=50" will slow down the speed of execution and might be helpful in studying the program. Don't forget to go back to "SPEED=255" if you want the computer to return to normal operating speed. The command "NOTRACE" will return the Apple to normal function. TRACE,NOTRACE AND SPEED commands may be used in the immediate or deferred mode (within a program).

We have created *"nested subroutines,"* in which there is a subroutine within a subroutine and each "RETURN" used, directs the program back to THE NEXT LINE after where that particular subroutine began!! Perhaps it would make it easier to think "subroutine #1," "RETURN #1," "subroutine #2," "RETURN #2" and add a REM to the particular line :

```
50 GOSUB 1000:REM GOSUB #1
1010 GOSUB 2000:REM
   GOSUB #2
1030 RETURN :REM RETURN #1
2020 RETURN:REM RETURN #2
```

POP ART

Thus, when the program is listed or studied, you can identify which GOSUB belongs to which RETURN. The REMS can be removed later if there is a shortage of memory, but at the beginning, this will help you keep track of things.

Now add line 2005:

2005POP

and run the program once more. This time, the RETURN sends the program to line 60 directly and not back to line 1020. The numbers printed will be 10,20,30,40,50,80,90, 60, and 70. Note that "100" was not printed. Line 1020 was ignored totally!! Study the flow of the program so that you understand why each command functions as it does.

The POP command caused the computer to "FORGET" the second RETURN and sends the program to the first RETURN. This enables the programmer to exit the NESTED SUBROUTINE and RETURN to the main program.

Why would you want to use POP?

NLLSQ

SCIENTIFIC SOFTWARE FOR THE APPLE II®

SOPHISTICATED NONLINEAR LEAST SQUARES

USING MARQUARDT OPTIMIZATION STRATEGY

Curve-fitting, parameter estimation, data-fitting- - - - all performed with convenience and speed on your own APPLE®

In using NLLSQ it is only necessary to provide simple one-line subroutines to fit data to equations like:

$y = A \exp(B + Cx) + D \exp(E + Fx)$

$y = A(x - x_0)^2 / [1 + B(x - x_0)^2]$

$y = A \cos(Bx + C) + Dx^E$

Almost any linear or nonlinear least squares problem can be handled, subject to limitations of memory and machine accuracy.

NLLSQ INCLUDES FEATURES LIMITED PREVIOUSLY TO MAIN-FRAME COMPUTING

Disk Storage of Results
Formatted Output
Standard Deviations in Parameters

Write for free, informative NLLSQ manual

NLLSQ . . . $125.00 + $3.00 handling

CET RESEARCH GROUP, LTD.
P. O. BOX 2029
Norman, OK 73070
(405) 360-5464

®Trademark of Apple Computer, Inc.

Whenever the conditions set up in the second (or nested) subroutine have been met, and your program no longer needs to stay there!! As an example, if the second subroutine is counting, the command might look like this: "IF N= 100 THEN POP: RETURN." POP allows the program to RETURN to the original Sub routine cleanly and smoothly.

Here is a diagram showing this:

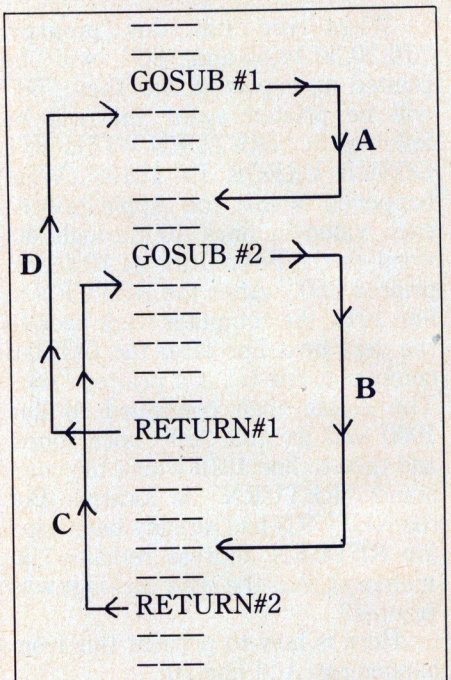


Fig 1. Without POP

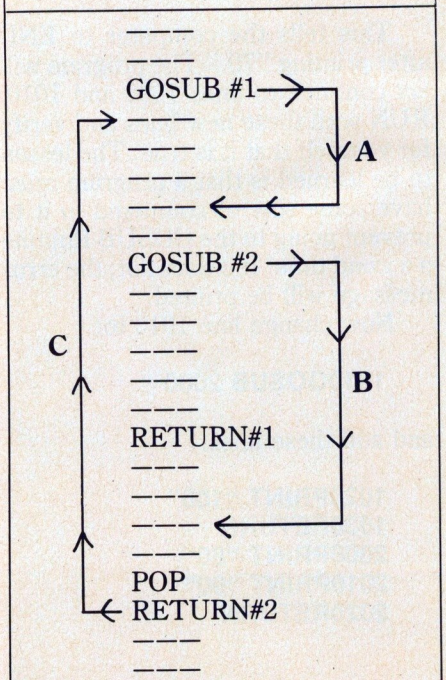


Fig 2 With POP

Here is what occurs without POP:

Line A is the path the computer follows to some line further along in the program, which then continues until the second GOSUB is found. Line B is the path taken by the second GOSUB command. When the Apple reaches RETURN #2, the program goes back to the line immediately following GOSUB #2, line C, continues till RETURN #1, and then back to the line after GOSUB #1, along path D.

With POP:

Lines A and B remain the same, but line C shows the path the program takes after the POP is executed.

In our example, the RETURN in line 1030 is the first (because it occurs earliest in the program and is the RETURN of the "OUTER" Sub) and the RETURN in line 2020 is the second. It is the RETURN of the "INNER" Sub. Thus POP when followed by a RETURN, sends the computer to the very next line of the earlier GOSUB command.

In the Apple there is a block of memory, called "THE STACK," that is used by the computer as you and I would use a note book! Whenever we wish to record some sort of information for future use, we write this data in our note books and when we later need these facts, we look them up. For example, think of the last time you went shopping. You prepared a list of what you wished to buy and consulted the list as you made your purchases. Our Apples do not go shopping, but they do have to remember many things as they operate.

Here is a small program that demonstrates this:

```
10  PRINT "ONE"
20  GOSUB 100
30  PRINT "THREE":END
100 PRINT "TWO"
110 RETURN
```

Line 10, of course, prints "ONE," and then the program goes to the subroutine at line 100. Here "TWO" is printed and then the program is told to RETURN. Suppose that lines 20 and 30 were not given numbers. How then would the computer "know" where to RETURN? If the Apple could talk, it would say, "WHERE TO?"

The program does have line numbers, so it branches back to very next line following the GOSUB command. Thus line number 20 must be "remembered" by the computer! This is done by recording this fact into the "Stack" which is the "note book," where the Apple keeps temporary information that will be needed later. The stack not only keeps track of RETURN addresses, but also the intermediate results of calculations, and is a very busy place. It is a block of memory starting at Hex address 100 which extends through Hex address 1FF, and consists of 256 decimal bytes, as a little calculation demonstrates.

Each 256 byte (decimal) block of memory in the Apple is given the name of "page." Addresses between 00 and FF (in Hex) are on page zero; the stack is called page one.

There are three important things to remember about the stack:

1. Stack memory is limited to only 256 bytes of data.

A capacity of 256 bytes appears to be quite large. Yet too much data will exceed the stack's storage capacity. This is why the "OUT OF MEMORY" error message will be given if the program contains too many nested "GOSUBS" (more than 25 levels deep), too many "FOR-THEN" loops (more than 10 levels deep), or nested parenthesis (more than 36 levels deep).

In reality, the computer has not necessarily exceeded its memory, only its capacity to store data in the stack; so the "OUT OF MEMORY" is not true. The error message should be, "STACK OVERFLOW."

A CALL 54915 command within an Applesoft program will start a machine language program which starts at Hex address $D683. This will clear up the "OUT OF MEMORY" error. You can examine this by first typing, "CALL-151," hitting return, and type, "D683L" and, of course, return again. The machine language of this subroutine will appear on your screen, and if you study the listing, you will learn something about assembly programming.

2. The stack does not actually store data as data, but only the address where the data can be found.

This is a more memory efficient method to store data....when the Apple needs this information, all it has to do is find where it was stored, and then look there. This greatly increases the practical applications of data storage and retrieval which are controlled by the stack.

3. The stack operates in a "FIRST IN, LAST OUT" mode.

The first piece of information stored by the stack goes to memory location 1FF. Subsequent pieces go to 1FE,1FD and so on. Consider location 1FF the bottom of the stack, and location 100 the top. Thus when information is added to the stack, the most recently added is closest to the top, and the oldest is at the bottom. Information is removed from the stack in the reverse order, from the top toward the bottom, or "LAST IN, FIRST OUT."

When a "GOSUB" command is executed, the line number containing this command is stored by the computer at an address "remembered" by the stack. The next time the command "RETURN" is found, the **top** address stored in the stack is used to direct the Apple to the appropriate line number, and then that information is removed from the stack, leaving the next address line to be used next.

Here is what the stack might look like:

```
TOP
   LINE NUMBER GOSUB #3
   LINE NUMBER GOSUB #2
   LINE NUMBER GOSUB #1
BOTTOM
```

The address of the first GOSUB is entered first and is at the bottom of the stack. As each additional line number is added, they are placed on top of the one added earlier. The information stored in the stack is removed in the reverse direction, i.e., from the top. This is what the stack would look like after the RETURN of GOSUB #3 is executed:

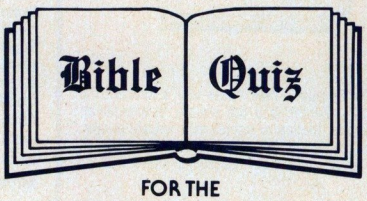
```
TOP
   LINE NUMBER GOSUB #2
   LINE NUMBER GOSUB #1
BOTTOM
```

The command POP can now be easily understood: It "ZAPS" the top line number of the stack! The line number that was second from the top, now becomes the the top line stored, and the next RETURN is directed to that portion of the program.

Perhaps, thinking "Nested subroutines" will help you understand POP. I certainly hope so, and also hope that you will try this command. It really is easy to use and a worthwhile addition to any programmer's bag of tricks.

I would like to thank Val Golding for making some suggestions which I have used in writing this article. Thanks, Val.

Sebastian Software
PRESENTS



FOR THE
APPLE II® COMPUTER
REQUIRES DOS 3.2 OR 3.3, APPLESOFT

PROGRAMS INCLUDE
- **BIBLE QUIZ**
- **ADD VERSES**
- **FILE MAINTENANCE**

Easily memorize verses of scripture with this unique system. Enter the verses you wish to remember into the data base using the **Add Verses** Program. The **Bible Quiz** Program will then formulate questions on all of the verses you have selected to be quizzed on. There are 5 different types of questions each verse may have with a variety of different ways to ask each type:

1. Reference a quoted verse.
2. Give the topic of a verse number.
3. Quote a numbered verse.
4. Pick the verse number in a given chapter for a particular subject.
5. Fill the blanks of a verse.

You may group your verses by Books of the Bible, topics, or whatever you wish.

Available now for only $29.95 postpaid from:

PROFESSIONAL COMPUTER SYSTEMS
318A LINCOLN CT.
BLOOMINGDALE, IL 60108

FUTRA COMPANY

(213) 328-8951 (800) 421-5006
TWX 910 349 6211
AGENFTRA TRNC

Mail Order: **Department C3**
P.O. Box 4380
Torrance, CA 90510-4380

Retail: **OMC 20695 S. Western Ave.**
Suite 124
Torrance, CA 90501
(213) 328-1760

apple computer
Sales and Service

APPLE VOICE COMMAND

- Easy to use - Interfaces with off-the-shelf software of programs may be written in BASIC, INTERGER-BASIC, APPLESOFT-BASIC, OR MACHINE CODE
- 40-word vocabulary with overlay feature to access additional vocabularies from disk
- Specially designed 16K ramcard (Cache) does not conflict with the original 48K nor with other additional expansion ramcards
- High accuracy (98%+)

In our endeavor to offer our customers the brightest stars in state-of-the-art products at the best value. Futra Company announces **"Shadow Vet"** by Scotts Instruments. The first true and functional voice command console unit developed for any microcomputer and available specifically for your Apple II. Simply speak any of up to 40 totally re-programmable commands and this device will translate it into the defined keyboard response you require for the program you are operating with. This means you can concentrate on the actual work at hand and not on the keyboard. Imagine controlling Visicalc by voice You can "GOTO" ay cell in the grid, or "RECALCULATE" the total spread sheet all by simple voice commands. The Shadow Vet is supplied with pre-processor, Cache interface board, software, noise cancelling headset/microphone and operators manual. Designed for the professional and business user, we at Futra think Shadow Vet is probably one of the best personal productivity devices designed for the Apple. Give us a call so you may speak to your Apple and make it work harder for you. **$889.00**

Video Terminals & Monitors

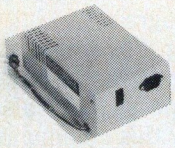
Zenith 12" $125

Sanyo:
- 9" Green Monitor

NEC:
- 12" Green Monitor
- 12" Color Monitor (Composite)
- 12" Color Monitor (RGB)

M & R ENTERPRISES

SUP 'R' FAN $42.50



SUP 'R' SWITCHER $219.00

Sup 'R' Mod	**$32.00**
RH Electronics Super Fan II	**$67.95**

MOUNTAIN COMPUTER

8 SLOT EXPANSION CHASSIS

with power supply

Increase your Apple capability and reduce the power drain on your Apple computer with the Expansion Chassis.

ROMPLUS	**$145**	Music System	**$355**
ROMWRITER	**141**	Introl System	**270**
APPLE CLOCK	**195**	X/10 Control	**160**
CPS Multi-function card			**$165**

EPSON PRINTERS

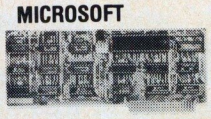
Still the best valued printer around. Friction and removable tractor, 9x9 to 18x18 dot matrix, logic seeking, with graftrax option built in.

MX-100 $ CALL $

•MX-80	•MX-80 FT.
•MX-70	•Graftrax
•Serial Int. Opt.	•2K Serial Int.
•Parallel Int.	•Par. Cable
$ $ CALL $ $	**$ $ CALL $ $**

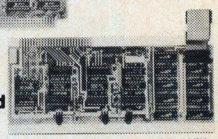
MX-80 Ribbon **$14.00**

MICROSOFT



$ CALL Z-80 Soft Card with CP/M

$ CALL 16K Ram Card



Open your Apple II compuer to a larger world. With the Z-80 softcard and 16K RAM card you can now run CP/M compatible software. Expand your memory for specific application, act as a firmware card and much more. If you add any boards to your Apple this year, these are the ones.

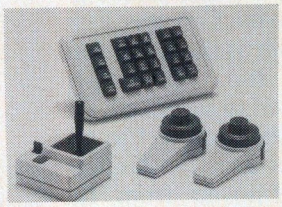
KONAN

Stop the floppy shuffle!
New Konan 5" Winchester Hard Disk Drive.
Introductory Pricing

$2299.00

THE KEYBOARD COMPANY

TKC



JOYSTICK II	**$ 42.95**
GAME PADDLES	**27.00**
NUMERIC KEYPAD	**129.95**

HAYES MICROMODEM II $303

The Micromodem II is a complete communication system for the Apple II

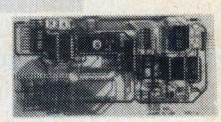
NOVATION APPLE CAT II $311

Buy the Apple Cat II, the modem that gorws with your needs. Find out about Novation's add-on features

VIDEX INC.

$245 VIDEOTERM

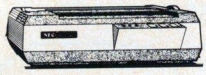
$122.95 KEYBOARD ENHANCER II



Manual Switch Plate	**$19**
Electronic Switch Plate	**32.50**
Micromodem II Firmware ROM	**25.00**
7 x 12 Character Set	**39.00**

NEC 8023A PRINTER

$539.00



The Futra Company now offers you a choice. The NEC 8023A Printer is a dot matrix, bi-directional, logic-seeking, friction or tractor feed, impact printer. At 100 characters per second you owe it to yourself to find about this value packed printer. Call us

ADVANCED LOGIC SYSTEM

Synergize Your Apple! The Synergizer permits use of the full range of CP/M software on Apple II and Apple III, including word processing, data base management, forecasting, accounting and more. It's an upgrade system all from one manufacturer that has been Apple Computer approved.

Synergizer Package Includes:

Z-CARD: Provides Z-80A microprocessor in the Apple with CP/M system and utilities.

SMARTERM: 80 column video display board with automatic 40 to 80 column switching from the keyboard and true shift key operation.

ADD-RAM: 16K RAM memory expansion for the Apple computer.

Synergizer Package $609.00

Items may be purchased separately
Z-Card **$218.00**
Smarterm .. **$309.00**
Add-RAM ... **$122.00**

VISTA A800 DISK CONTROLLER

$481

The Vista A800 8" floppy disk controller offers you a true advantage. With the A800 and 8" drives you can now have up to a full 1 megabyte per disk drive.

MicroSci 8" Floppy Disk Drive ..... **$Call**

MISC. HARDWARE

CALL FOR PRICES

SSM Microcomputer Products
AIO-II, AIO, APIO, ASIO A488, AKB1, AXB1

Advanced Business Technology
Keypad A or B, Barwand, Softkey

High Technology
Apple Juice (Reserve Power Supply) **$325**

ALF: 9 and 3 Voice Music Card **$ CALL**

Orange Micro
Grappler Epson MX—80/100 & Grappler NEC 8023A **$149.00**

APPLE SOFTWARE — GAMES

Automated Simulations

Starfleet Orion	19.00
Invasion Orion	19.00
Temple of Apshai	29.00
Hellfire Warrior	29.00
Star Warrior	29.00
Crush, Crumble & Chomp	25.00
the Dragon's Eye	19.00
Jabbertalky	25.00
Sorcerer of Siva	25.00

Avante-Garde:

Hi-Res Computer Golf	26.00
Race for Midnight	26.00
Hi-Res Secrets	85.00

Broderbund Software

Apple Panic	23.96
Space Quarks	26.95
Genetic Drift	26.95
Track Attack	26.95
Red Alert	26.95
Midnight Magic	29.70
Arcade Machine	35.96
Snoggle	

Budgeco
Raster Blaster

Continental Software
LA Land Monopoly
Hyperspace Wars
3-D Skiing
Home Money Minder

Cal Pacific Computer
Ultima

Dakin 5/Level-10
Alkemstone Game
Kaves of Karkhan
Dragon Fire
Rings of Saturn

Gebeli Software Inc
Firebird

Inovative Designs
Pool 1.5
Shuffel Board

35.96	Trick Shot	35.95	Gobbler
	Microsoft Consumer Products		**Personal Software**
23.96	Olympic Decathalon	23.96	Microchess 2.0
	Typing Tutor	19.96	Checker King
26.95	Microsoft Adventure	26.95	Gammon Gambler
26.95	**Muse Software**		Bridge Partner
22.45	Robot Wars	31.96	Monty Plays Monopoly
29.70	Three Mile Island	35.95	Monty Plays Scrabble
	Best of Muse	35.95	**Riverbank Software**
31.96	Castle Wolfenstein	23.96	Int'l Grand Prix
	ABM	19.96	**Sierra Software**
33.95	**On-Line Systems**		Alien Lander
44.95	Wizzard & Princess	29.65	Retro Ball
44.95	H/R Football	35.95	Space Adventure
35.95	Pegasus II	23.96	**Sirius Software, Inc.**
	Misle Defense	23.96	Cops & robbers
23.96	Cranston Manor	31.45	Spoch
	Hi-Res Soccer	26.95	Orbitron
	Ulysses & Golden Fleece	31.45	Gamma Gobblins
27.96	Soft Porn Adventure	23.96	Sneakers
26.95			

	Phantom Five	26.95	
	Space Eggs	26.95	
22.45	Gorgon	31.96	
	Star Cruiser	22.45	
22.45	Cyber Strike	35.95	
22.45	Autobahn	26.95	
22.45	Pulsar II	26.95	
22.45	**Strategic Simulation**		
31.45	Computer Bismark	42.00	
31.45	Computer Ambush	42.00	
	Operation Apocalypse	42.00	
23.96	Shattered Alliance	42.00	
	Computer Quarterback	27.00	
19.96	Computer Conflict	27.00	
25.95	Warp Factor	27.00	
25.95	Cartels & Cutthroats	27.00	
	President Elect	27.00	
	Computer Baseball	27.00	
31.45	Battle of Shiloh	27.00	
27.96	Tigers in the Snow	27.00	
26.95	**Sub Logic**		
26.95	Flight Simulator	26.80	
23.96			

DATA SUPPLIES

Silvermans Diskette Sleeves $6.95
Store and protect your diskettes/**Pkg. of 10**

Scotch Products: For reliable data recording use the best diskette for the money. With reinforced center!

Box of 10 - 5¼" storage carton ... **$27.00**
Scotch Head Cleaning Kit **$26.00**

CALIFORNIA COMPUTER SYSTEMS

CALL

7712A Serial Sync	7470A A/D Converter
7710A Asynch serial card	7720A Parallel
784B Arithmetic process	7728 A Centronics
7440 Program Timer	**Calendar Clock Module**

BOOKS

AOSBORNE:

Business Systems Buyers Guide	6.75
Practical BASIC Prog. (Apple II)	14.40
Some Common BASIC Prog.	13.45
Apple II User's Guide	15.00
CP/M User's Guide	11.00
CBASIC User's Guide	13.50
Science & Engineering Prog.	14.80
6502 Assembly Language Prog.	15.30
6502 assembly Language Prog.	11.70
Z-80 Assembly Language Prog.	15.30
Word Star Made Easy	7.25
Management Information Source	
The Power of Visicalc	9.05

GENERAL APPLE SOFTWARE

Personal Software Visicalc, Visitrend/Visiplot, Visidex **$ CALL** Visiterm, Visiplot

Microsoft		**Art SCI**	
Cobol 80	**$562**	Magic Window	**$Call**
Fortran 80	**175**	**Information Unlimited**	
ALDS	**95**	Easy writer 40 col.	**79.96**
TASC	**149**	Easy Mailer 40 Col	**62.95**
Micropro		Easy Mover 40 Col	**44.95**
Wordstar	**$235**	**Software Publising Corp.**	
Datastar	**236**	Personal Filing Systems	**76.00**
Mail Merge	**112**	Personal Report Systems	**85.50**
Spellstar	**168**	**Stoneware**	
Calcstar	**156**	DB Master	**$179**
Supersort I	**159**	DB Master Utility Pack	**$89**
Peachtree Software			
Gen'l Ledger			**$180**
Acct. Rec./Pay, Payroll, Inventory			**225**

Please specify 40 or 80 col. version

PROMETHEUS

VERSACARD $CALL

It might be the only Apple Peripheral interface you need. The Prometheus Versacard is a parallel, serial, real time clock, and BSR interface. It's fully compatible with CP/M and Apple Pascal.

PADDLE ADDAPPLE $29.00

Game I/O over loaded? Don't open your Apple. Get the two-plug Paddle-Addapple. Switch selectable and reasonably priced.

TERMS: Shipping: add 3% for product shipped within continental USA via UPS surface (minimum $6.00). **If the order placed is prepaid with U.S. funds in the form of check or money order, a Total Charge of $6.00 for shipping is all you pay within the continental USA via UPS surface.** Allow 14 working days for personal and company checks to clear. Credit card charges limited to $1000. No COD's. FPO, APO or orders outside continental USA call or write for shipping charges or add 20% to purchase price (any difference will be refunded). California residents add 6% sales tax. All offers subject to change or withdrawal without notice.

*CPM is a registered trademark of Digital Research.

EXPLORING THE WORLD OF APPLE:

Get the Most From a Diskette

By Clay Ruth

Philosophy

THE introduction of DOS 3.3 gave Apple users a whopping 23% increase in user-available storage per disk. Here's a way to further increase DOS 3.3 space by as much as 12%, depending on your needs.

Each DOS 3.3 diskette is formatted into 560 sectors (16 sectors on each of 35 tracks), but the user only has access to 496 of them. This is because DOS itself reserves 48 sectors, and the diskette catalog ties up 15 more. Let's take a look at these "unavailable" sectors and see what we can do with them.

First of all, DOS doesn't make full use of those 48 sectors it reserves! Sectors 5 thru F of track 2 are unused and are yours for the taking. You have to free them up in the track bit map of the Volume Table of Contents (VTOC). That's all there is to it! The means of doing this is described later in this article.

While we're looking at DOS, doesn't it seem rather wasteful to have a copy of it on every one of those diskettes in your library? Sure, it is convenient to be able to boot up any disk you have, but sometimes that just isn't necessary. For example, you may have software on one diskette that creates text files on another diskette. If you are using the second disk only for data, it certainly doesn't need a copy of DOS! You will never have occasion to boot up a data-only disk, so you might as well reclaim 47 of DOS's 48 sectors! You can't take track 0, sector 0, for reasons I'll get into later.

Then there's the catalog. It occupies all 16 sectors of Track 17 ($11). Sector 0 is reserved for the VTOC, while the remaining 15 sectors each contain 7 catalog entries. This makes room for 105 files which is a lot more than most of us ever put on one disk. To get this many files on a DOS 3.3 diskette would require the average file size to be less than 5 sectors! We can take back several sectors of the catalog track, so long as we preserve the VTOC sector and one sector for each 7 files we anticipate putting on the diskette. This requires changing both the VTOC and one catalog sector, and a minor modification to DOS.

SUGGESTIONS

These procedures require a general disk access utility, such as "Disk Zap" on A.P.P.L.E.'s Diskpak 6B-16. The utility you use must be able to read a specified sector, allow you to modify it, and write it back to the disk. Several good utility programs are on the market, so rather than re-invent the utility, I will tell you which sectors to change and recommend that you purchase one of these readily-available products.

I do not recommend that you start right out and add space to a disk that is already full of data. Starting with a newly-initialized disk, make the modifications, then move several large files to it to test its capacity. When you are satisfied that everything works properly and you feel comfortable using the utility, you will be ready to tackle your already-filled diskettes. Even then, it is wise to first make a backup copy, just in case you accidentally clobber the wrong sector.

PROCEDURE

Step 1:

Initialize a blank diskette in the usual manner, as per the Apple DOS manual, and use your chosen utility program to read Track 17 ($11), sector 0. This is the VTOC. It contains track bit maps for all 35 tracks. As described in the DOS manual, they look quite complicated, but they are really quite simple once you get to know them.

The VTOC for a newly-initialized DOS 3.3 diskette appears in Fig.1A. The track bit maps start at Byte $38 of the VTOC sector and continue through byte $C3. Four consecutive bytes represent one track, but only the first two bytes are used; the third and fourth always contain zeros. These first two bytes of each track's bit map tell DOS which sectors of that track are available for writing files. Since one byte contains eight binary bits, those two bytes contain 16 bits...one for each sector of the track. A bit can only be 1 or 0; DOS uses these values to represent a sector available or occupied, respectively. When additional sectors are required by a file, DOS examines the VTOC, searching for remaining one-bits in the track bit maps. When it finds one, it writes data to the corresponding sector and changes the bit to 0. Later, if the file is deleted, DOS changes all bits corresponding to that file's sectors back to 1's. But we don't see the 1's and 0's; we just see hexadecimal numbers, so we need an easy way to convert binary 1's and 0's to hexadecimal before we can safely modify the VTOC.

I find it convenient to look at those first two bytes of each track's map as a group of four hexadecimal digits, each digit representing four sectors of the track, as shown in the bit chart in Fig.2. The first digit represents sectors F,E,D, and C. The second digit is sectors B, A, 9, and 8; the third is 7, 6, 5 and 4, and the fourth digit is sectors 3, 2, 1 and 0.

GET THE MOST FROM A DISKETTE

Looking at Fig.1A, you will find that most of the track bit maps for a newly-initialized diskette contain FF FF, indicating empty tracks. The track $12 bit map (at bytes $80-81) contains 3F FF; this is because the greeting program occupies sectors F and E of that track. Since tracks are usually filled starting at sector F and working down, the most commonly seen digits in a partially filled diskette are F, 7, 3, 1 and 0, representing 4, 3, 2, 1 and 0 available sectors in that digit's group of four. This may not seem obvious at first, but should make more sense after you've completed step 2. A full diskette's track bit maps are all full of zeros.

In Fig.1A, four bit maps read 00 00, indicating fully occupied tracks. These full tracks appear in bytes $38-39 (track 0), $3C-3D (track 1), $40-41 (track 2), and $7C-7D (track $11). These are the locations we are about to change.

Step 2:

For each affected track, decide which sectors you want to make available and figure out what hexadecimal value represents a 1-bit in the position corresponding to each free-up sector. It is easiest to do this by making a bit chart like the one shown in Fig.2. Fill in the "Bits Set" row by placing a 1 in each column corresponding to a sector you want to free up. Place zeros under all sectors

For A.P.P.L.E. Members Only

Woz Talks -

90 MINUTE VHS VIDEO TAPE

from the A.P.P.L.E. meeting, Oct., 1981

1. Early Microcomputers
2. Development of the Apple I and][
3. A look ahead
4. Question and answer session

Member Price $27.50

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055

Delivery subject to production schedule

Washington state residents add 6.6% sales tax.

All foreign orders (incl. Canada) will require additional postage and shipping.

Figure 1A

Volume Table of Contents (VTOC) from Track 17 ($11), Sector 0 of a newly-initialized diskette with a 2-sector greeting program.

Directory begins on Track $11, Sector $0F. This is a DOS 3.3 diskette.

Volume Number is $FE (254 decimal)

```
9000- 04 11 0F 03 00 00 FE 00
9008- 00 00 00 00 00 00 00 00
9010- 00 00 00 00 00 00 00 00
9018- 00 00 00 00 00 00 00 00
```

A sector of a Track/Sector List contains up to $7A (122) track/sector pairs

```
9020- 00 00 00 00 00 00 00 7A
9028- 00 00 00 00 00 00 00 00
```

Last wrote Track $12. Stepping by 1. $23 (35) tracks; each is $10 (16) sectors. A sector has $0100 (256) data bytes.

```
9030- 12 01 00 00 23 10 00 01
9038-  Trk 0   00 00  Trk 1
9040-  Trk 2   00 00  Trk 3
9048- FF FF 00 00 FF FF 00 00
9050- FF FF 00 00 FF FF 00 00
```

```
9058- FF FF 00 00 FF FF 00 00
9060- FF FF 00 00 FF FF 00 00
9068- FF FF 00 00 FF FF 00 00
9070- FF FF 00 00 FF FF 00 00
9078-  Trk $10   00 00  Trk$11
9080-  Trk$12   00 00  Trk$13
9088- FF FF 00 00 FF FF 00 00
9090- FF FF 00 00 FF FF 00 00
9098- FF FF 00 00 FF FF 00 00
90A0- FF FF 00 00 FF FF 00 00
90A8- FF FF 00 00 FF FF 00 00
90B0- FF FF 00 00 FF FF 00 00
90B8- FF FF 00 00 FF FF 00 00
90C0-  Trk$22   00 00 00 00 00 00
90C8- 00 00 00 00 00 00 00 00
90D0- 00 00 00 00 00 00 00 00
90D8- 00 00 00 00 00 00 00 00
90E0- 00 00 00 00 00 00 00 00
90E8- 00 00 00 00 00 00 00 00
90F0- 00 00 00 00 00 00 00 00
90F8- 00 00 00 00 00 00 00 00
```

Track bit maps for 35 tracks numbered $00 through $22 (0 through 34)

BUSINESS AUTHORS WANTED

(APPLE AND IBM P.C.)

If you have written Business Software for the Apple Computer, or the IBM personal computer, Westware, Inc., would like to hear from you.

Westware, Inc., offers national advertising, experienced documentation writing, and software support. If you are interested in nationally marketing your business product, write:

WESTWARE SOFTWARE, INC.
2455 S. W. 4th Avenue, Suite #2
Ontario, OR 97914

	Byte 1		Byte 2	
	Digit 1	Digit 2	Digit 3	Digit 4
Sector No.	F E D C	B A 9 8	7 6 5 4	3 2 1 0
Bit Value	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1
Bits Set	_ _ _ _	_ _ _ _	_ _ _ _	_ _ _ _
Hex Digit	_ _ _ _	_ _ _ _	_ _ _ _	_ _ _ _

Fig. 2 — Bit Chart for converting binary to hexadecimal (Step 2)

that are to remain protected. Then, for each digit (one at a time!), add up all the bit values corresponding to the 1's you just filled in, remembering that 10=A, 11=B, 12=C, 13=D, 14=E, and 15=F. Note that if you are freeing up all four sectors corresponding to one digit, you place 1's in all four columns. The digit value is then 8+4+2+1=15=F, so you place an F in that digit.

To reclaim part of the catalog, determine how many files (maximum) you expect to put on the disk. Divide by seven and round up to the next higher whole number to determine how many sectors your catalog really needs. Construct the bit chart and place a 0 under sector 0 (you don't want to clobber the VTOC!). Starting at sector F and proceeding to the right, place zeros in as many consecutive places as you need sectors for your catalog. Then fill in the rest of the places with 1's. Add up the bit values corresponding to the 1's for each digit. Depending on how many files you anticipated, digit 1 will be 7, 3, 1, or 0, corresponding to 7, 14, 21 and 28 possible files, respectively. Digit 2 will probably be F, but it may be 7, 3, 1 or 0 if you were to expect more than 28 files. Digit 3 will probably be F, and digit 4 will be E. Place the resultant values in bytes $7C-7D of the VTOC. A modified VTOC with room for 14 files appears in Fig. 1B. If you later try to create more files than you've allowed room for, you will get a "DISK FULL" message, even though free sectors remain.

Now let's reclaim the unused portion of Track 2. Look at bytes $40-41 of your VTOC. They contain 00 00. Change them to FF E0. Checking the bit chart, you'll find this is what you get if you free up sectors F through 5. This has been done to the VTOC in Fig.1B.

If you have decided to sacrifice DOS, change VTOC bytes $40-41 to FF FF. Do the same to bytes $3C-3D. Then change bytes $38-39 to FF FE (Sector 0 of track 0 can't be used for files, so we set the rightmost bit to 0). The resultant VTOC appears in Fig.1C.

Figure 1B

Same VTOC after completing Steps 1 and 2, if DOS is to be kept. In this example, room was reserved for 14 catalog entries (file names).

```
9000- 04 11 0F 03 00 00 FE 00
9008- 00 00 00 00 00 00 00 00
9010- 00 00 00 00 00 00 00 00
9018- 00 00 00 00 00 00 00 00
9020- 00 00 00 00 00 00 00 7A
9028- 00 00 00 00 00 00 00 00
9030- 12 01 00 00 23 10 00 01
9038- 00 00 00 00 00 00 00 00
9040- FF E0 00 00 FF FF 00 00
9048- FF FF 00 00 FF FF 00 00
9050- FF FF 00 00 FF FF 00 00
9058- FF FF 00 00 FF FF 00 00
9060- FF FF 00 00 FF FF 00 00
9068- FF FF 00 00 FF FF 00 00
9070- FF FF 00 00 FF FF 00 00
9078- FF FF 00 00 3F FE 00 00
9080- 3F FF 00 00 FF FF 00 00
9088- FF FF 00 00 FF FF 00 00
9090- FF FF 00 00 FF FF 00 00
9098- FF FF 00 00 FF FF 00 00
90A0- FF FF 00 00 FF FF 00 00
90A8- FF FF 00 00 FF FF 00 00
90B0- FF FF 00 00 FF FF 00 00
90B8- FF FF 00 00 FF FF 00 00
90C0- FF FF 00 00 00 00 00 00
90C8- 00 00 00 00 00 00 00 00
90D0- 00 00 00 00 00 00 00 00
90D8- 00 00 00 00 00 00 00 00
90E0- 00 00 00 00 00 00 00 00
90E8- 00 00 00 00 00 00 00 00
90F0- 00 00 00 00 00 00 00 00
90F8- 00 00 00 00 00 00 00 00
```

Figure 1C

Same VTOC after completing Steps 1 and 2, if DOS is to be sacrificed.

```
9000- 04 11 0F 03 00 00 FE 00
9008- 00 00 00 00 00 00 00 00
9010- 00 00 00 00 00 00 00 00
9018- 00 00 00 00 00 00 00 00
9020- 00 00 00 00 00 00 00 7A
9028- 00 00 00 00 00 00 00 00
9030- 12 01 00 00 23 10 00 01
9038- FF FE 00 00 FF FF 00 00
9040- FF FF 00 00 FF FF 00 00
9048- FF FF 00 00 FF FF 00 00
9050- FF FF 00 00 FF FF 00 00
9058- FF FF 00 00 FF FF 00 00
9060- FF FF 00 00 FF FF 00 00
9068- FF FF 00 00 FF FF 00 00
9070- FF FF 00 00 FF FF 00 00
9078- FF FF 00 00 3F FE 00 00
9080- 3F FF 00 00 FF FF 00 00
9088- FF FF 00 00 FF FF 00 00
9090- FF FF 00 00 FF FF 00 00
9098- FF FF 00 00 FF FF 00 00
90A0- FF FF 00 00 FF FF 00 00
90A8- FF FF 00 00 FF FF 00 00
90B0- FF FF 00 00 FF FF 00 00
90B8- FF FF 00 00 FF FF 00 00
90C0- FF FF 00 00 00 00 00 00
90C8- 00 00 00 00 00 00 00 00
90D0- 00 00 00 00 00 00 00 00
90D8- 00 00 00 00 00 00 00 00
90E0- 00 00 00 00 00 00 00 00
90E8- 00 00 00 00 00 00 00 00
90F0- 00 00 00 00 00 00 00 00
90F8- 00 00 00 00 00 00 00 00
```

Note: If DOS is sacrified, Steps 5, 6A and 6B must done on a different diskette. Step 6C must be done on the DOS-less diskette. The 'different' diskette must be one that is certain to be booted before the DOS-less diskette is accessed.

GET THE MOST FROM A DISKETTE

Step 3:

You are done modifying the VTOC. Write it back to track $11, Sector 0.

Step 4:

Look at the bit chart you made for the catalog (track $11) and determine the lowest-numbered sector you reserved for catalog entries. If, for example, you reserved three sectors (room for 21 files), this would be sector D. Read this sector. You will find it full of zeros, except bytes 01 and 02. Byte 01 contains $11, and byte 02 contains the number of the next-lower sector. Change them both to 00. This tells DOS that the catalog ends here, so it won't try to interpret your new data sectors as catalog entries. Write the sector back where it came from.

Step 5:

Read track 2, sector 1 of the diskette that, in normal usage, will be booted up before the modified diskette is written to. If you are *not* sacrificing DOS, this will be the same diskette you have been modifying so far. However, if DOS *has* been eliminated, a different diskette (presumably one containing software that accesses files on your newly-modified diskette), must be booted before the modified diskette is of any use. This bootable diskette is the one we are working with in steps 5, 6A and 6B. Consider the "Alternate DOS Modification" following step 6C, then perform steps 5 through 6B only if you decide to modify DOS directly on the diskette. Make sure you have a backup copy if it contains valuable data!

You will find a $69 at byte $92. Change the $69 to $A9, and write the sector back to the diskette. This changes a DOS internal instruction that would normally prevent it from ever examining track $11 to see if it had any free sectors! DOS normally searches tracks $12 through $22, then $10 through $00, never bothering to check out track $11. This one byte modification makes it start with track $11 on every pass.

Step 6:

If you are not sacrificing DOS, you can skip steps 6A through 6C because they are only necessary if files are to be written to track 0. Actually, writing to track 0 is no problem, but standard DOS won't let you read the data back! DOS reads a

Listing 1 — For Track 0, Sector 6 of bootstrap disk

Memory Size Bytes E0-F1

```
32K  E0:  C8  B1  42  D0  07  88  AD  BB  75  4C  08  71  88  B1  42  4C  14  71
48K  E0:  C8  B1  42  D0  07  88  AD  BB  B5  4C  08  B1  88  B1  42  4C  14  B1
```

Listing 2

```
:ASM

                        1    * DOS INTERCEPT ROUTINE *
                        2    *
                        3    *
                        4            ORG   $BCE0
                        5    *
BCE0: C8                6            INY                ;Increment index to sector posn
BCE1: B1 42             7            LDA   ($42),Y      ;Get sector nbr from T/S list
BCE3: D0 07             8            BNE   $BCEC        ;GOTO $BCEC if sector not zero
BCE5: 88                9            DEY                ;Sector is zero; restore orig index

BCE6: AD BB B5          10           LDA   $B5BB        ;This is the instruction covered
                        11   * when we installed the intercept vector at $B105
BCE9: 4C 08 B1          12           JMP   $B108        ;GOTO instruction after intercept
                        13   * vector. (Declare 'END OF DATA')
BCEC: 88                14           DEY                ;Restore original index
BCED: B1 42             15           LDA   ($42),Y      ;Reload with track nbr to be read
BCEF: 4C 14 B1          16           JMP   $B114        ;GOTO where DOS has accepted track
                        17   * and gets ready to read file's next sector.
```


For A.P.P.L.E. Members Only

DICE stands for Dan's I/O Control Enhancer, which permits the easy direct entry of lower case characters, along with the "inaccessible" Apple characters _\[and their control counterparts. It will work with any Apple][equipped with Ap.L Case or any other lower case display chip. It is available separately, or in the following combinations:

DICE DISKETTE $12.50

Ap. L Case 1* Chip plus DICE: $46.00

Ap. L Case 2† Chip plus DICE: $37.50

Ap. L Case is the chip you need to display lower case characters on your standard screen.

* for Rev 6 and older Apples † for Rev 7 and later Apples

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055

Dealer inquiries invited.
VISA/M-C welcome.

Washington residents must add sales 6.6% sales tax.

All foreign orders (incl. Canada) will require additional postage and shipping.

file by reading sectors in the sequence that they appear in the file's track/sector list. When it encounters a track 0 reference, it aborts the read and gives an END OF DATA error. To get around this, we must intercept DOS where it detects the track 0 reference and make it test the sector reference too. If the sector is nonzero, we return to where DOS would have gone in a normal read sequence. But if the sector reference is also zero, we go to the END OF DATA error, just as DOS originally intended to do. Now you know why we can't free up track 0, sector 0!

Step 6A:

Fortunately, there is unused space within DOS where we can place the intercept routine. Read track 2, sector 0 of the disk that will be booted before the modified data disk is read (the same disk you used in Step 5). Look at byte $07. If it contains a $35, you are working with a master diskette. Any other value indicates a slave diskette; the memory size is indicated in table A. Change bytes $05 through $07 by entering the appropriate values from listing 3. Write it back to the disk. This modification intercepts DOS where it has found a track 0 reference and sends it to the intercept routine.

Table A: Significance of Byte 07 of Track 2, Sector 0

If Byte Is	Then the diskette is
$35	Master (see note)
$75	32K Slave
$B5	48K Slave

Note: $35 can also mean 16K Slave, but this is an impractical memory size for DOS, so I assume anyone reading this article has more. If you are modifying DOS on a master diskette, use the portion of Listings 1 and 3 corresponding to your system's memory size.

Step 6B:

Read track 0, sector 6 of the same disk you used in steps 5 and 6A. Starting with byte $E0 of this sector, enter the appropriate portion of listing 1. This sets up the machine code shown in listing 2 (for 48K) or its equivalent for your memory size. Then write the sector back to the diskette.

Listing 3 — For Track 2, Sector 0 of bootstrap disk

Memory Size	Bytes 05-07
32K	05: 4C E0 7C
48K	05: 4C E0 BC

Note: All bytes other than 05, 06 and 07 must remain as they were.

Step 6C:

Now that we are making use of everything except track 0, sector 0, let's give that forgotten sector something to do. This sector is the first one loaded in when booting DOS, and it is executed immediately after it is read. We can turn this to our advantage, even though we don't plan to boot DOS!

Suppose you mistakenly try to boot a disk on which DOS has been overwritten by files. The results would be unpredicatable! We can alleviate this possibility by setting up this special sector to print a little reminder such as *To Access These Files, Boot Disk Volume 123* or *Try a Different Disk, Dummy* or whatever suits your fancy. You can print up to 189 characters (including spaces, carriage returns, etc) at your choice of VTAB positions by doing the following.

AMPERGRAPH

This is the graphics utility package that you have been awaiting. AMPERGRAPH adds twenty-two Applesoft commands that allow effortless generation of professional-looking plots of scientific or financial data. All of the necessary scaling and screen formatting is accomplished with just a few Applesoft lines.

Unlike most other plotting systems for the Apple II which are stand-alone systems, the AMPERGRAPH utility provides extended BASIC graphics language commands that you can use directly in your own Applesoft programs. The additional commands are &SCALE, &LIMIT, &AXES, &GRID, &FRAME, &LOG X, &LOG Y, &LABEL AXES, &LABEL, &VLABEL, &CENTER LABEL, &CENTER VLABEL, &DRAW, &PENUP,&CROSS, &OPEN SQUARE, &CLOSED SQUARE, &OPEN CIRCLE, &CLOSED CIRCLE, &ERROR BARS, &DUMP (for Silentype printer), and &*DUMP (to link with AMPERDUMP, see below).

AMPERGRAPH uses the Applesoft ampersand machine language jump vector to link to a relocatable 9K routine which normally resides above the second high-resolution graphics page in the Apple II. The system requires an Apple II Plus (or Apple II with language card) with 48K and DOS 3.3.

AMPERDUMP

This is a high-resolution graphics dump utility which was written specifically to take advantage of the graphics features of the Epson MX-80 and MX-100 printers (MX-80 must have the Graftrax conversion). AMPERDUMP offers many features which are not available in other graphics dump routines:

* Three horizontal magnifications (2.33, 4.66 and 6.99 inches wide)
* Nine vertical magnifications (0.91, 1.77, 2.64, 3.78, 4.25, 4.45, 5.31, 5.87 and 7.99 inches high)
* Horizontal and vertical magnifications can be specified independently to obtain 27 different plot size formats
* Normal/Inverse dumps
* Easy to use
* Adjustable horizontal tab
* Fast
* Compatible with AMPERGRAPH
* Relocatable

AMPERGRAPH and AMPERDUMP are available from your dealer for $30.00 each, or order direct. Include $1.50 for shipping and handling; Wisconsin residents add 4% sales tax.

Madwest Software
Dept. C
P.O. Box 9822
Madison, WI 53715

GET THE MOST FROM A DISKETTE

With your disk utility running, read track 0, sector 0 of the modified data diskette. You will find that byte 0 contains 01. It must remain that way, or the disk controller card will reject the data when trying to boot. Make this sector look like listing 4, substituting your message (in ASCII) where I show ?? ?? —— ?? ?? (Note! The number of bytes in your message is not important, so long as it does not exceed 189 characters. It must be followed immediately by the 9 bytes shown).

Table 7 on Page 15 of the Apple II Reference Manual will help you ascertain the correct hex codes for your ASCII message. Bear in mind that a carriage return is really a Ctrl-M ($8D), and an additional line feed (to skip to the next line without affecting the horizontal cursor position) is a Ctrl-J ($8A).

Now write the data back to track 0, sector 0. It doesn't matter if there are old bytes remaining after the ones you entered; they will be ignored. Try to boot the diskette. You should see your message displayed, followed by a BASIC prompt. If your message doesn't use up all 189 bytes, you can add some ASCII linefeed characters ($8A) to the end of it, to move the BASIC prompt farther down the screen.

Step 7:

If you sacrificed DOS, delete the greeting program from the modified data diskette. This will free up the sectors it once occupied.

Alternate DOS Modification

The DOS modifications made in steps 5, 6A and 6B require you to know that a particular diskette will always be booted up before you attempt to access the files on your DOS-less diskette. But sometimes you just can't be sure! So here is an alternate method. It POKEs in the DOS changes when the controlling BASIC program begins executing.

Listing 5 is a five line Applesoft program that will modify the DOS in RAM just as though you had performed steps 5, 6A and 6B. The program automatically corrects for your system's memory size, except for the DATA statement. See the note in listing 5 for these corrections.

The numbers in line 0 are simply the decimal equivalents of the hex bytes in listing 1. Line 1 determines your system's memory size and computes the proper place to POKE the DATA values into your DOS. Line 2 actually POKE's them in (equivalent to step 6B), and line 3 POKE's in the intercept vector (as in step 6A). Line 4 does a POKE equivalent to step 5. Line 5 is needed only if you have other DATA statements in your pro-

Listing 4 — Modification to Track 0, Sector 0

```
                        1
                        2  *
0800: 01                3
0801: 20 93 FE          4
0804: 20 89 FE          5
0807: 4C 2D 08          6
080A: 68                7
080B: 85 19             8
080D: 68                9
080E: 85 1A             10
0810: A0 00             11

0812: E6 19             12
0814: D0 02             13

0816: E6 1A             14
0818: B1 19             15
081A: C9 88             16
081C: F0 08             17
081E: 09 80             18
0820: 20 ED FD          19
0823: 4C 10 08          20
0826: A5 1A             21

0828: 48                22
0829: A5 19             23
082B: 48                24
082C: 60                25
082D: 20 58 FC          26
0830: A9 05             27
0832: 85 25             28
0834: 20 22 FC          29
0837: 20 0A 08          30
083A: BF BF AD
083D: AD AD BF
0840: BF                31
                        32
0841: 88                33
                        34
0842: A6 2B             35
0844: BD 88 C0          36
0847: 4C 00 E0          37
```

```
          ORG   $0800

          HEX   01          ;Make it acceptable to controller
          JSR   $FE93       ;Set video as output device
          JSR   $FE89       ;Set keyboard as input device
          JMP   $082D       ;Skip over the printing routine
          PLA               ;Get low byte of calling address
          STA   $19         ;Save it for future reference
          PLA               ;Pull high byte of calling address
          STA   $1A         ;Save it too
          LDY   #$0         ;Zero offset for in direct load at $818

          INC   $19         ;Increment low order ref. byte
          BNE   $0818       ;Skip next step if no page boundary

          INC   $1A         ;Increment high order ref. byte
          LDA   ($19),Y     ;Load from current ref. address
          CMP   #$88        ;Check for end of msg flag
          BEQ   $0826       ;If so, branch to exit routine
          ORA   #$80        ;Set high bit for normal video
          JSR   $FDED       ;Print the character
          JMP   $0810       ;Advance to next data byte
          LDA   $1A         ;Get high byte of last ref. address

          PHA               ;Save it on the stack
          LDA   $19         ;Get low byte of last ref. address
          PHA               ;Push it on stack
          RTS               ;Return to address after end flag
          JSR   $FC58       ;HOME
          LDA   #$05        ;VTAB 6
          STA   $25         ;Store at cursor vertical
          JSR   $FC22       ;Execute the VTAB
          JSR   $080A       ;Print the following text

          ASC   "??---??"
* Enter your message in this area, up to 189 bytes
          HEX   88          ;Message end flag
* Following addresses depend on msg length
          LDX   $2B         ;Fetch slot number of boot
          LDA   $C088,X     ;Turn off disk drive motor
          JMP   $E000       ;Cold start BASIC
```

gram (it keeps the new DATA statement from being read when you really wanted to read something else.

Other Modifications

On most drives, it is possible to add a 36th track, producing an additional 16 sectors. Since there are some drives that can not handle this extra track, I do not recommend using it. However, the interested reader can find information on this in *Nibble*, Vol. 2, No. 6.

Those of you who do not yet have DOS 3.3 are probably wondering if these modifications will work on earlier versions of DOS. It can be done, but the numbers are different, and the customized message display (step 6C), doesn't work. My advice to you is simply upgrade to DOS 3.3!

CREDITS

I wish to thank Call -A.P.P.L.E. for recommending the book (Beneath Apple DOS). This book is every bit as good as Val says it is! Without it, I would never have succeeded in making tracks 0 and $11 useable.

CAUTION

The procedures described in this article are intended for use only on diskettes that were initialized by an Apple Computer using DOS 3.3. Most commercial software, etc., use custom operating systems which may already make use of the areas described in this article. Many are protected such that they may not be accessed by a normal disk utility, therefore you should not attempt to modify any diskette that you did not initialize yourself, unless you are 100% certain that these sectors are unused.

Once a diskette has been modified, it is wise to label it distinctively so you will not attempt to modify it again at a later date. The author and A.P.P.L.E. are not responsible for any loss of data resulting from the application or misapplication of these procedures.

Listing 5 — Alternate method of modifying DOS

Include the following at the start of your Applesoft program that will access files on your modified data-only (DOS-less) diskette:

```
0   DATA 200,177,66,208,7,136,173,
      187,181,76,8,177,136,177,66,
      76,20,177: REM For 32k use:
0   DATA 200,177,66,208,7,136,173
      ,187,117,76,8,113,136,177,66
      ,76,20,113
1  S = 256 *  PEEK (987) + 1504
2   RESTORE : FOR X = S TO S + 17:
      READ Y: POKE X,Y: NEXT
3   POKE S - 3035,76: POKE S - 303
      4,224: POKE S - 3033, PEEK (987) + 5
4   POKE S - 2638,169
5   DEL 0,4: REM Use this line
only if your program contains
other DATA statements
```


SOFTWARE SOLUTIONS

MASTER SCHEDULER: Sales Persons & Managers Rejoice! Keep track of when to contact people, when to review people. Classify people or accounts your way. Schedule up to 20 calendar days for yourself. Recall people by classification. Recall who to contact today or for a week. Recall who to review today. Print-out a name and address list or names, address, and phone list. Print-out mailing lables too. Handles up to 179 separate people or accounts.
Requires: Apple II, 48k RAM, Applesoft in ROM, 2 disk drives-DOS 3.3, and 80 column printer $129.95

PRODUCT PROFIT ANALYSIS: Be More Profitable! Easily analyze how a change in any Cost Element or Price effect your break-even volume, profit per unit, % profit per unit, and total profit at various volumes of sales. Screen and Print-out results.
Requires: Apple II, 48k RAM, Applesoft in ROM, 1 disk drive-DOS 3.3, and 80 column printer with compressed characters $34.95

EXCLUSIVE BUSINESS SOFTWARE

JOB COSTING & PRICING ANALYSIS: Save Time Analyzing or Writing Proposals and in Negotiations! Input data includes: Mfg. Labor, Tech. Eng. Labor, PA Eng. Labor, Material, Other Direct Costs, FCCM, Eng. O/H, Mfg. O/H, G&A, Profit or Fee. Up to 1,500 different cost items. *Average* Labor, Material and Other Direct Costs escalations. Easily see how any change effects the bottom line. Full Print-out of all input data and summary. Develop multiple negotiation positions in minutes. Save and reuse proposals or modify.
Requires: Same as Product Profit Analysis $69.95

CIPVIRR: Compound Interest, Present Value, Internal Rate of Return. *Invest Smartly* using this program to solve these types of problems. Simple to follow with great examples.
Requires: Same as Product Profit Analysis, but no printer $49.95

VISA Dealer Inquiries Invited MasterCharge

Phone Orders: (408) 438-2433
SOFTWARE SOLUTIONS
9124 Hwy. 17
Scotts Valley, CA 95066

Please specify printer.
CA Residents add 6.5% sales tax.
Foreign orders add $5.00 postage and handling per product.

FINANCIAL MANAGEMENT SYSTEM II

A HOME AND BUSINESS ACCOUNTING SYSTEM
FAST - EASY TO USE

Enter an entire month's CHECKING, CHARGE CARD, and CASH accounts in just a few minutes using your own personalized macro lists. INSTANT ENTRY ERROR CORRECTION. Audit all files by Code and Month with year-to-date totals. PERFECT FOR TAX ACCOUNTING.

- SELF PROMPTING, ERROR AVOIDING ENTRY SYSTEM. 1 to 3 KEYSTROKE entries with AUTOMATIC DATE, CODING and NUMBERING.
- SCREEN and PRINTER routines for listing files, balance, reconcile, search, and audit reports. Configure system to match almost any printer.
- BUDGET MANAGER to plan, review and balance your budget, generate COMPLETE reports for any 1 – 12 month period.
- SEARCH all files with specific and expanded searches.
- ACCOUNT AUDITOR totals all files by Code and any 1 – 12 month period with year-to-date totals.
- Enter your own Item and Code Macros, up to 100 each.
- 48K with ROM APPLESOFT and DISK required. SINGLE or DUAL DISK compatable. (printer optional)
- **PRICE $39.95**

***** NOW AVAILABLE *****

CHECK WRITER II for FINANCIAL MANAGEMENT SYSTEM II. Print your checks directly from FMS II files.

- **PRICE $19.95 – ($15.00 if ordered with FMS II)**

GROCERY LIST

A USEFUL HOUSEHOLD PROGRAM DESIGNED TO ORGANIZE SUPERMARKET SHOPPING

Shoppers will INSTANTLY be able to use this easy, self-prompting program. Scan a file of up to 500 USER DEFINED ITEMS. Choose those needed with a single keystroke. Then print a shopping list ORGANIZED BY TABLE NUMBER, SECTION, or four letter code such as "DAIRY", "BAKE", or "DELI".

- 32 – 48K with disk and printer required, (APPLESOFT)
- **PRICE $19.95**

COMPUTERIZED MANAGEMENT SYSTEMS
by D R JARVIS COMPUTING INC.
1039 Cadiz Dr. - Simi, CA 93065
PHONE (805) 526-0151
Check, VISA or MASTER CARD accepted. *DEALER INQUIRES INVITED*

GET FREE SOFTWARE FOR YOUR APPLE !

HOW? JUST ORDER ONY OF THE ITEMS BELOW, AND SELECT YOUR FREE SOFTWARE FROM THE BONUS SOFTWARE SECTION, USING THE FOLLOWING RULE: FOR THE FIRST $100.00 WORTH OF MERCHANDISE ORDERED TAKE 1 ITEM; FOR THE NEXT $200.00 WORTH OF MERCHANDISE ORDERED TAKE ANOTHER ITEM; FOR THE NEXT $300.00 TAKE A THIRD ITEM, ETC. ALL AT NO COST.

HARDWARE BY APPLE

APPLE II PLUS, 48k	1199
DISK DRIVE+CONTROLLER (3.3)	535
DISK DRIVE only	455
Language System w. Pascal	397
Silentype Printer & Interface	360
Integer or Applesoft Firmware Card	159
Graphics Tablet	645
Parallel Printer Interface Card	149
Hi-Speed Serial Card	155
Centronics Parallel Intfce.	175

HARDWARE BY OTHERS

HAYES MICROMODEM II	300
HAYES SMART MODEM	239
HAYES S100 MODEM	339
VIDEX VIDEOTERM 80 W. GRAPHICS	275
MICROSOFT Z80 SOFTCARD	299
MICROSOFT 16K RAMCARD	159
CORVUS 10MB HARD DISK	4750
SSM A10 SERIAL/PARALLEL A&T	189
MICRO-SCI Disk & Controller	495
TYMAC DOUBLE DOS 3.2/3.3	35

VIDEO MONITORS

Leedex-Video-100 12" B&W w/Cable	139
Leedex 12" Green w/Cable	165
Leedex 13" COLOR MONITOR & Cable	399
SUP-R-TERM RF MODULATOR	29

HARDWARE BY MOUNTAIN COMPUTER

Clock/Calendar Card	239
A/D & D/A Interface	319
Expansion Chassis	555
ROMplus Card	135
Mark Sense Card Reader	995
CPS Multifunction Bd.	239

SOFTWARE FOR APPLE

APPLE FORTRAN	159
APPLE PILOT	125
DOS 3.3	50
DOS TOOL KIT	65
APPLE PLOT	59
D. J. REPORTER	45
D. J. NEWS	45
PORTFOLIO	45
SHELL GAMES	25
ELEMENTARY DEAR APPLE	25

SOFTWARE BY OTHERS

APPLE FORTRAN by MICROSOFT	159
APPLE BASIC COMPILER by MICROSOFT	315
APPLE COBOL by MICROSOFT	599
VISICALC 3.3	169
VISIPLOT	155
VISIPLOT/VISITREND	199
VISIDEX	169
CCA DATA MGT.	79
DB MASTER by STONEWARE	189
DATACAPTURE 4.0	55
Z – TERM	65
ON-LINE APPLESOFT COMPILER	89

SOFTWARE BY PEACHTREE

GEN. LEDGER	219
A/R	219
A/P	219
PAYROLL	219
INVENTORY	219
MAIL LIST	219

WORD PROCESSING SOFTWARE FOR APPLE

PEN-ULTIMATE	235
WORD STAR	245
EZ WRITER Prof. Sys.	195
EZ WRITER	89
MUSE SUPER TEXT II	139
APPLE-WRITER	69
PROGRAMMA APPLE PIE 2.0	110
MAGIC WAND	345
WORDPOWER	50

EPSON PRINTERS

MX - 70 w/Graftrax	415
MX - 80	515
MX - 80 FT	615
MX - 80 w. GRAFTRAX	575
MX - 80 FT w. GRAFTRAX	675
MX - 100 FT w. GRAFTRAX	775
APPLE PAR. INTFCE (for all Epson)	75
MX -70/80 FRICTION FEED Adaptor	75

OTHER PRINTERS

IDS 445 w. GRAPHICS + 2K Buff.	750
IDS 460 w. GRAPHICS	899
IDS 460	825
IDS 560	1099
IDS 560 w. GRAPHICS	1150
CENTRONICS 737	699
CENTRONICS 739	799
CCS Centronics Par. Intfce & Cable	135
NEC SPINWRITER 5510 RO	2795
C. ITOH 25 CPS DAISYWHEEL	1750
C. ITOH 45 CPS DAISY WHEEL	2025
WATANABE MI-PLOT PLOTTER	1150
DIABLO 630 DAISY w. P. Wheel & Rib.	2350

BONUS SOFTWARE SECTION!

Let us acquaint you with MESSAGE-MAKING SOFTWARE. Just place the disk in the APPLE, enter the text, and colorful, dynamic messages appear on the screens of TV sets connected to the computer. Use the software to broadcast messages on TV screens in schools, hospitals, factories, store window, exhibit booths, etc. The following program is our latest release:

SUPER MESSAGE: Creates messages in full-page "chunks". Each message allows statements of mixed typestyles, typesizes and colors, in mixed upper and lower case. Styles range from regular APPLE characters, up to double-size, double-width characters with a heavy, bold font. Six colors may be used for each different typestyle. Vertical and horizontal centering are available, and word-wrap is automatic. Users can chain pages together to make multi-page messages. Pages can be advanced manually or automatically. Multi-page messages can be stored to disc or recalled instantly.
REQUIRES 48K & ROM APPLESOFT..... $50.

APPLE PLOTS YOUR DATA & KEEPS YOUR RECORDS TOO! APPLE DATA GRAPH 2.1: Plots up to 3 superimposed curves on the Hi-res Screen both X & Y axes dimensioned. Each curve consists of up to 120 pieces of data. Graphs can be stored to disc and recalled immediately for updating. Up to 100 graphs can be stored on the same disc. Great for Stock-market Charting. Business Management, and Classroom instruction!
REQUIRES 48K & ROM APPLESOFT..... $35

APPLE RECORD MANAGER: Allows complete files to be brought into memory so that record searches and manipulations are instantaneous. Records within any file can contain up to 20 fields, with user-defined headings. Information can be string or numeric. User's can browse thru files using page-forward, page-backward or random-search commands. Records can easily be searched, altered or sorted at will. Files can be stored on the same drive as the master program, or on another, if a second drive is available. Records of files can be printed, if desired. Additional modules coming are a STATISTICS INTERFACE, CHECKBOOK, MAILING LIST & DATA-ENTRY.
REQUIRES 48K & ROM APPLESOFT..... $40

APPLE LITERATURE DATABASE: allows rapid retrieval (via keywords) of references from total APPLE literature thru 1980, on 5.25" disk. Each entry in the data base consists of the article, author-name, periodical-name, date of issue, & page nos. The database is intended to support large magazine files which would require lengthy manual searching to recover information. Annual updates will be available.
REQUIRES 48K, ROM APPLESOFT..... $60

WORDPOWER: is a simple, powerful, low cost, line-oriented word-processor program. It offers a fast machine language FIND & REPLACE. Text can be listed to screen or printer, with or without line-numbers. Lower-case adaptors are supported. You can merge files, move groups of lines, and easily add, change, or delete lines. WORDPOWER can be used to create and maintain EXEC files. It can also be used as a rapid, unstructured, information-storage and retrieval system via its rapid search capabilities.
REQUIRES: 48K, ROM APPLESOFT..... $50

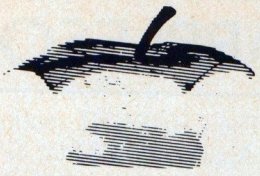
LABELMAKER: allows users to quickly create address labels. A given label may be generated in any quantity from 1 to 32767. Space is allowed on labels for a personal and company name, but the space is automatically closed up if only a personal name is entered. Space is also allowed for foreign countries. The program can also generate labels for price-tags, part numbers and mail-messages such as "RUSH", "FRAGILE", etc. A self-incrementing feature allows theatre-tickets to be produced, with a date, and numbers running from a000 to z999. An editor is provided for editing labels prior to printing. All labels may be saved to disk for instant recall.
REQUIRES 48K & ROM APPLESOFT..... $35

TO ORDER: Use phone or mail. We accept VISA, MASTERCARD, COD's, personal checks & money orders. Add 4% for credit card. Customer pays handling on COD orders, Foreign orders must be in American Dollars & include 10% for handling. Connecticut residents add 7.5% sales tax.

CONNECTICUT INFORMATION SYSTEMS CO.
218 Huntington Road, Bridgeport, CT 06608 (203) 579-0472

FROM THE VERY CORE OF APPLE:

Applesoft's CHARGET Routine



By Cornelis Bongers/Call —A.P.P.L.E. Staff Writer

1. Introduction

THIS article gives an explanation of the CHARGET routine of Applesoft. Attention shall be paid to the functions of the CHARGET routine and its structure. The article is oriented towards the beginning machine language programmer. However, it is assumed that the reader is already reasonably familiar with the 6502 instruction set.

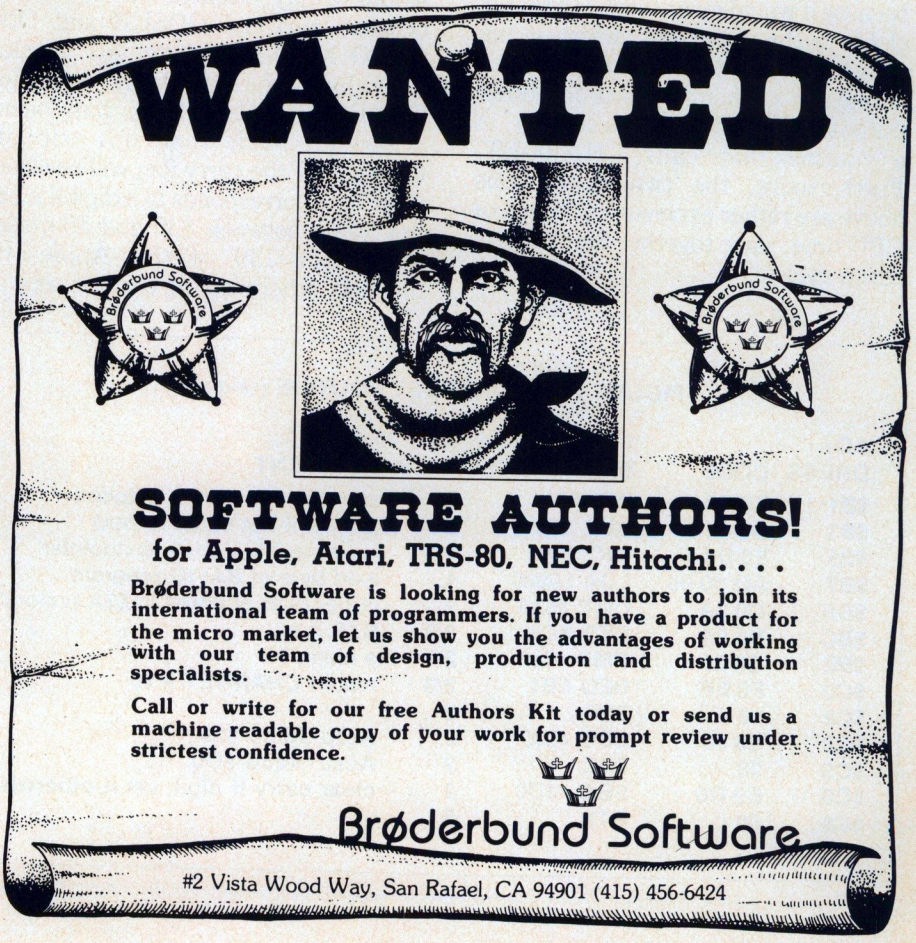
CHARGET is one of Applesoft's most frequently used routines. In order to get a good understanding of CHARGET it is first necessary to say a few words about the environment in which it operates. On the Apple II plus, the Applesoft interpreter resides in the upper part of memory ($D000-$F7FF). If you key in a program, this happens under supervision of the interpreter. The interpreter reads each line from the keyboard, translates it to "tokenized" form (which means that all keywords are being replaced by one byte codes with a value > = 128), removes all blanks in the line (except those between quotes or in DATA or REM statements) and stores the line in memory.

Both the interpreter and the BASIC program are nothing more than a set of hex data stored somewhere in memory. The difference is that the sequence of hex values which make up the interpreter are meaningful to the 6502, because these values form a machine language program. The BASIC program, on the other hand, cannot (directly) be executed by the 6502, because it is not a machine language program but rather a sequence of lines of the "coded" English.

The execution of the BASIC program is the task of the interpreter. The interpreter "knows" what sequences of machine language instructions have to be executed if, for instance, the BASIC instruction A = A+1 (represented by the hex values $41,$D0,$41, $C8,$31), is encountered. The interpreter breaks up the task, A = A+1 into a number of sub-tasks. All these sub-tasks can be performed by machine language subroutines which are already present in the interpreter.

2. The Functions of Charget

After these introductionary remarks, we return to the CHARGET routine. The CHARGET routine is, in a sense, in the same way related to the interpreter as the address and data line are being related to the 6502. If the 6502 wants to read the contents of a particular memory cell, it puts the address of that cell on the address line and fetches the contents of the cell from the data line. In case the interpreter wants to read the next byte of a BASIC program it calls the CHARGET routine. CHARGET returns with the next byte loaded in the Accumulator.



APPLESOFT'S CHARGET ROUTINE

The CHARGET routine resides in the Applesoft ROM (at $F10B) and it has a length of 24 bytes. When Applesoft is started, the routine is copied to the zero page (at $B1). The reason is that CHARGET is self-modifying, or in other words, it changes itself when it is executed. Usually, self-modifying routines are considered to be bad programming practice, but for CHARGET there are good reasons, which I shall discuss below.

The CHARGET routine is listed in figure 1. The most important memory locations used by the routine are $B8 and $B9. These locations contain a pointer, called the textpointer, which points to the current byte of the BASIC program. At each call to CHARGET, the pointer in $B8, $B9 is incremented first, so that it points to the next byte. Then the byte is fetched from memory. Normally, one would use for this purpose the instructions:

```
LDY #$00
LDA ($B8),Y
```

However, here this is not efficient because it would mean that CHARGET clobbers the Y-register, which is unattractive since it would imply that the Y-register cannot be used for other purposes (for instance as a counter) in a routine that calls CHARGET. Furthermore, the two instructions above are rather expensive in terms of CPU time, since together they require seven cycles.

Therefore another method has been applied which consists of putting the storage area of the textpointer directly behind a LDA instruction. As can be seen from figure 1, the opcode $AD for LDA resides at $B7, just before the address of the textpointer, so each time $B8, $B9 is incremented, the LDA instruction will automatically load the next byte of the BASIC program in the Accumulator. Note that CHARGET can also be used to load the current byte in the Accumulator with the JSR $B7 instruction.

After the execution of the LDA instruction, the most important task of CHARGET has been performed. However, CHARGET has more functions. When retrieving bytes from a BASIC program, it is often necessary to check whether a byte corresponds to [the ASCII representation of] a digit. For example, when Applesoft finds the token for GOTO somewhere in a BASIC program, it knows that a number must follow this token. So the next byte has to be checked for being a digit. This of course can be done within the GOTO handler, but this would imply duplication of code, because also the GOSUB and ON GOTO handlers require this check. Therefore, the check is done in CHARGET. The check has been organized such that on exit of CHARGET the carry flag will be clear if the byte corresponds to a digit (i.e., a hex value between $30 and $39; see pages 138–139 of the Applesoft manual), otherwise the carry is set.

FIGURE 1 THE CHARGET ROUTINE (original)

ADDRESS	CODE	MNEMONICS	CYCLES	COMMENT
$B1	E6 B8	INC $B8	5	increment low byte textpointer
$B3	D0 02	BNE $B7	2/3	branch if not equal to zero
$B5	E6 B9	INC $B9	5	increment high byte textpointer
$B7	AD 00 00	LDA $0000	4	load byte of BASIC program
$BA	C0 3A	CMP #$3A	2	larger than or equal to $3A (colon)?
$BC	B0 0A	BCS $C8	2/3	RTS if so
$BE	C9 20	CMP #$20	2	a blank ?
$C0	F0 EF	BEQ $B1	2/3	restart CHARGET if so
$C2	38	SEC	2	
$C3	E9 30	SBC #$30	2	Accu=Accu-$30
$C5	38	SEC	2	Accu=Accu-$D0
$C6	E9 D0	SBC #$D0	2	clear carry if digit, set it otherwise
$C8	60	RTS	6	

Another check that often must be performed by Applesoft is the end of statement/line check. An end of a statement is reached if the current byte is a colon (hex: $3A) and an end of a line if the current byte is zero. The colon is used to separate multiple statements on a line and is inserted by the user. A zero signals that the end of a BASIC line is reached and is added by Applesoft when the line is keyed in. The CHARGET routine returns with the zero flag set if the retrieved byte equals $3A or $00, else the zero flag is clear.

Finally, the CHARGET routine skips blanks.

Let us next examine how CHARGET performs these checks. After the Accumulator has been loaded with a byte of the BASIC program, this byte is compared with $3A, which is the ASCII representation of a colon (see figure 1). The CMP instruction has the effect that the carry will be set if the byte in the Accumulator (further to be called Accu-byte), is larger than or equal to $3A, otherwise the carry is cleared. Furthermore, the zero flag will be set if the Accu-byte equals $3A, otherwise it is cleared.

The next instruction is a branch "on carry set" instruction, and as will be clear, the branch is taken if the Accu-byte is larger than or equal to $3A (because in this case the carry is set). Control is then returned to the calling program or subroutine via the RTS instruction. Note that upon execution of the RTS via the BCS instruction both the zero flag and the carry flag have their proper value.

If the branch is not taken, the Accu-byte is compared with $20 which is the ASCII representation of a blank. In case the Accu-byte equals $20, a branch to the start of the routine is executed. This has the effect that blanks will simply be ignored by the CHARGET routine.

Now we come to the most obscure part of CHARGET. By means of the final four instructions, CHARGET determines whether the Accu-byte has a value less than $3A. What has to be done is to clear the carry if the value of the byte is equal to or larger than $30 (i.e., the digit 0), and to set it otherwise. In addition, the Accu-byte must not be clobbered and furthermore, the zero flag must be set if the Accu-byte equals zero.

It will be clear that the instruction CMP #$30 cannot be used for the digit check, because this instruction will set the carry, rather than clear it, if the Accu-byte corresponds to a digit. (The complement carry instruction is not implemented on the 6502).

So another method has to be used. This method consists of first subtracting $30 from the Accu-byte. This has the effect that the Accumulator assumes a value between $30-$30=0 and $39-$30=$9 if the Accu-byte was a digit, otherwise the Accumulator will get a value between $0-$30=$D0 and $2F-$30=$FF. Next the value $D0 is subtracted from the Accumulator. As a consequence, the Accumulator will be restored to its original value because, neglecting the overflow, $30+$D0=$00. Furthermore, if the Accu-byte corresponded to a digit, the last subtraction will clear the carry, because in the other case, the carry is set. Note that if the Accu-byte was equal to zero, the last subtraction will put the Accumulator equal to zero again, which sets the zero flag.

3. Speeding up CHARGET

After having described what the functions of CHARGET are and how CHARGET works, it is interesting to see whether its coding can be improved. Of course, this must happen under the conditions that: first, CHARGET must still support all the functions mentioned above and second, the available number of bytes (i.e., 24) must not be exceeded. Our goal is not to reduce the length of the routine, but rather to decrease its execution time. Before we start working on this, it might be useful to mention that no great practical results can be expected. The execution time of the CHARGET routine (including the JSR and the RTS) takes (on the average) about 32 cycles. Even if we reduce this by 10% (which would be quite substantial), the execution time of a BASIC program of 10K without loops, branch instructions and subroutines would be reduced by about .030 seconds only. Nevertheless, it is an interesting machine language programming exercise to determine whether CHARGET can be made more efficient and the "tricks" that will be applied may be useful in other —more time critical— programs.

When looking at the routine, the first part seems rather efficient, but the second part (starting with SEC) looks promising. There are four instructions, whose only function is to clear the carry if the Accu-byte corresponds to a digit, which seems a little overdone.

As mentioned before, we cannot use the CMP instruction to clear the carry in case the Accu-byte corresponds to a digit. This is due to the fact that the carry is cleared if the Accu-byte is smaller than the byte with which it is compared. However, if we take the complement of the Accu-byte with respect to a suitable byte value first, the use of the CMP instruction will lead to the desired result. Suppose, we take the complement with respect $FF, which can be done by executing the 6502's EOR #$FF instruction. The series $0-$39 will then be transformed to the series $FF-$C6, as is illustrated by figure 2.

FIGURE 2 EFFECT OF THE EOR #$FF INSTRUCTION

```
$00 --------> $FF
$01 --------> $FE
  .
  .
  .
$20 --------> $DF
  .
  .
  .
$2F --------> $D0
$30 --------> $CF
$31 --------> $CE
  .
  .
  .
$38 --------> $C7
$39 --------> $C6
```

Now, if we insert a CMP #$D0 instruction behind the EOR #$FF instruction, the carry will be cleared if the Accu-byte corresponded to a digit, else the carry will be set. After the CMP instruction, the Accumulator must be restored to its original value, which can be done by executing the EOR #$FF instruction once again.

The modified CHARGET routine is listed in figure 3. The length of this routine is the same as the length of the original one. The number of instructions, however, is reduced by one. As a result, the execution time of the routine will be reduced by about two microseconds if the Accu-byte is smaller than $3A but not equal to $20 (digits and control characters), since SEC, SBC #, CMP # and EOR # all take two cycles, and one cycle lasts about one microsecond.

Unfortunately, most BASIC programs contain relatively few digits and control characters, so improving the "digit" check may not be the best thing to do. A better improvement is streamlining the most frequently taken path through the routine. Since most bytes of a BASIC program are tokens or characters, the BCS instruction at $BC will more frequently lead to a branch to the RTS instruction than to the execution of the next instruction. A branch that is taken consumes three cycles, whereas a branch that is not taken takes only two cycles.

FIGURE 3 THE CHARGET ROUTINE (improvement)

ADDRESS	MNEMONICS	CYCLES	COMMENT
$B1	INC $B8	5	
$B3	BNE $B7	2/3	
$B5	INC $B9	5	
$B7	LDA $0000	4	
$BA	CMP #$3A	2	
$BC	BCS $C8	2	
$BE	CMP #$20	2	
$C0	BEQ $B1	2/3	
$C2	EOR #$FF	2	take complement with respect to $FF
$C4	CMP #$D0	2	clear carry if digit, set it otherwise
$C6	EOR #$FF	2	restore Accu
$C8	RTS	6	

APPLESOFT'S CHARGET ROUTINE

Therefore, it is obviously more efficient to replace the BCS instruction by a BCC instruction, which implies that the latter instruction must be followed by a RTS. This increases the length of the routine with one byte. Thus we have to save a byte at the blank/digit check. This can be done by inserting a SBC #$2F instruction directly behind the RTS and making this instruction the target of the BCC branch (see figure 4). The SBC #$2F instruction corresponds to the SBC #$30 instruction in the original CHARGET routine. Note that use is made of the fact that the value of the carry is known when this instruction is executed (the carry is clear) and that therefore the SEC instruction can be eliminated.

Next must be determined whether the Accu-byte was equal to a blank. Hereto, we have to compare the Accu-byte with $20-$30 = $F0 (rather than $20). If the Accu-byte equals $F0, the CHARGET routine is restarted. The rest of the routine is identical to the original CHARGET routine. The result of these changes is that one cycle is saved on every byte that is equal to or larger than $3A. Furthermore, if the Accu-byte is smaller than $3A but not equal to $20, one cycle is lost because of the branch that is taken, but two cycles are saved because one SEC instruction has been removed. This results in the saving of one cycle. Finally, if the Accu-byte corresponds to a blank, three cycles are lost. Therefore, the total number of saved cycles is:

1*f1 + 1*f2 - 3*f3

where

f1 = the relative frequency of bytes > $3A (characters and tokens)
f2 = the relative frequency of bytes < $3A but unequal to $20 (digits and control characters)
f3 = the relative frequency of bytes equal to $20 (blanks)

We can neglect the final term, because a tokenized BASIC program will contain virtually no blanks. So, we may conclude that the CHARGET routine listed in figure 4 is more efficient than the one listed in figure 3 as long as f1 > f2, which usually will be the case.

4. The Blank Check

Finally, some attention shall be paid to the blank check, because this check leads to problems now and then. For instance, if one wishes to put $A equal to the string bbITEM (b denotes a blank) with the INPUT statement, the interpreter will simply refuse to do this (unless the text is put between quotes, which is not a very convenient solution). Instead A$ is set to ITEM and the two leading blanks are ignored. One way to prevent this is to eliminate the blank check in CHARGET. Although this seems at first sight a little drastic, some experimentation showed that many programs run without problems if the blank check is removed. Also, inputting BASIC lines from the keyboard causes no problems, because superfluous blanks in BASIC lines are not "removed" by CHARGET but by some separate instructions of the Applesoft interpreter (at $D574 and $D5AB). However, problems may arise if a program contains DATA statements. When lines are entered via the keyboard and the interpreter encounters a DATA statement, the blank check is disabled, which has the effect that all text following the keyword DATA (until the next colon or end of line sign) will be stored exactly as the user inputs it. Thus, when typing DATAb1, the blank in front of the 1 will be stored in memory. Now, when executing the instruction READ A, the interpreter will first search for the DATA statement and if this statement is found, it starts converting the text directly behind the DATA keyword to a number. However, in our example, a blank is inserted behind "DATA" and since CHARGET doesn't skip blank lines anymore, this blank is returned to the interpreter, which will subsequently generate a SYNTAX ERROR, because it expects to receive a digit.

In order to avoid these difficulties, it is therefore better to disable the blank check (when necessary) only temporarily. Mr. Kurvers of the Erasmus University, with whom I discussed this matter, suggested to use the statements POKE 191,255:INPUT A$:POKE 191,32 if text has to be inputted with leading blanks. The first poke sets location $BF to $FF which, as can be seen from figure 1, has the effect that the branch at $C0 will never be taken because the Accu-byte will always be smaller than $FF. By means of the second poke, the blank check is enabled again.

Note that if instead of a string, a numeric value is inputted by means of this method (with INPUT A), a RE-ENTER message will be displayed if the number is preceded by one or more blanks. This will also be the case if one or more blanks are inputted between the digits or after the last digit of the number.

This is the extent of my knowledge of CHARGET. If anyone has additional suggestions or comments, I would appreciate hearing about them.

Cornelis Bongers
Erasmus University
Postbus 1738
3000 DR Rotterdam
The Netherlands

FIGURE 4 THE CHARGET ROUTINE (improvement 2)

ADDRESS	MNEMONICS	CYCLES	COMMENT
$B1	INC $B8	5	
$B3	BNE $B7	2/3	
$B5	INC $B9	5	
$B7	LDA $0000	4	
$BA	CMP #$3A	2	larger than or equal to $3A?
$BC	BCC $BF	2/3	branch if not
$BE	RTS	6	RTS if so
$BF	SBC #$2F	2	Accu=Accu-$30
$C1	CMP #$F0	2	is it a blank?
$C3	BEQ $B1	2/3	restart CHARGET if so
$C5	SEC	2	Accu=Accu-$D0
$C6	SBC #$D0	2	clear carry if digit, set it otherwise
$C8	RTS	6	

COLLEGE BOARD SAT* PREP SERIES

TRS-80, APPLE, PET, OSI, ATARI, CP/M, PDP-11

Each program confronts the user with a virtually limitless series of questions and answers. Each is based on past exams and presents material on the same level of difficulty and in the same form used in the S.A.T. Scoring is provided in accordance with the formula used by College Boards.

S.A.T., P.S.A.T., N.M.S.Q.T. — Educator Edition set includes 25 programs covering Vocabulary, Word Relationships, Reading Comprehension, Sentence Completion, and Mathematics. **Independent tests of S.A.T. series performance show a mean total increase of 70 points in students' scores.** Price **$229.95**

GRADUATE RECORD EXAM Series — Educator Edition includes 28 programs covering Vocabulary, Word Relationships, Reading Comprehension, Sentence Completion, Mathematics, Analytical Reasoning and Logical Diagrams. Price **$289.95**

COMPETENCY PROFICIENCY EXAM PREP SERIES

This comprehensive set of programs consists of simulated exam modules, a thorough diagnostic package, and a complete set of instructional programs. It is designed to teach concepts and operations, provide drill and practice and assess achievement levels through pre and post testing. The Competency Exam Preparation Series provides a structured, sequential, curriculum encompassing mathematical, reading and writing instruction.

This program is designed for individual student use or use in a classroom setting. Programs provide optional printer capability covering worksheet generation and performance monitoring. C.E.P.S. are available in three software formats.

Special editions available for California Proficiency Assessment Test and New York Regents Competency Tests. Call for Prices.

M.I.T. Logo for Apple $179.95

(Includes **Alice In Logoland**)

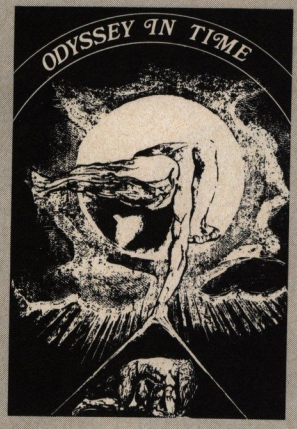
Odyssey In Time

This spectacular adventure game adds a new dimension of excitement and complexity to **Time Traveler.**

Odyssey In Time includes all the challenges of **Time Traveler** plus 10 additional eras. Each game is different and may be interrupted and saved at any point for later play. **$39.95**

Time Traveler

The best of the adventure games. Confronts the player with complex decision situations and the demand for real time action. Using the Time Machine, players face a challenging series of historical environments. To succeed you must build alliances and struggle with the ruling power. Each game is unique. **$24.95**



Isaac Newton + F.G. Newton

Perhaps the most fascinating and valuable educational game ever devised - **ISAAC NEWTON** challenges the players (1-4) to assemble evidence and discern the underlying "Laws of Nature" that have produced this evidence. **ISAAC NEWTON** is an inductive game that allows players to intervene actively by proposing experiments to determine if new data conform to the "Laws of Nature" in question. Players may set the level of difficulty from simple to fiendishly complex.

Full Graphics Newton. This version of Isaac Newton presents all data in graphic form. Because data is graphic rather than symbolic, this game is suitable for very young children. Players may select difficulty levels challenging to the most skilled adults. **$49.95**

☆ *NEW* ☆

☆MICRO-DEUTSCH☆

Micro-Deutsch set includes 24 grammar lessons, covering all material of an introductory German course. Four test units also included. Grammar lessons use substitution transformation drills, item ordering, translations and verb drills. Drill vocabulary based on frequency lists. Suitable for use with any high school or college textbook. Extensively field tested at SUNY Stony Brook. Available for Apple II and PET/CBM. (PET version includes a special foreign language character chip.) Also available soon: MICRO-FRANCAIS, MICRO-ESPAÑOL, MICRO-IVRIT, MICRO-YIDDISH, MICRO-CHINESE, MICRO-JAPANESE. **$179.95**

☆ *NEW* ☆

☆Pythagoras and The Dragon☆

Mathematics in a fantasy game context. Based on **The Sword of Zedek**, **Pythagoras and The Dragon** introduces Pythagoras as a mentor to the player. When called on for aid, Pythagoras poses math questions, and depending on the speed and accuracy of the player response, confers secret information. With Pythagoras as an ally, the quest to overthrow Ra, The Master of Evil, assumes a new dimension of complexity. Depending on the level chosen, problems range from arithmetic through plane geometry.

32K **$39.95**

Free Bonus with purchase of $300.00 or more: Applesoft Tutor Series

Krell Software Corp. has no official ties with the College Entrance Examination Board or the Educational Testing Service. Krell is, however, a supplier of products to the E.T.S.

PROGRAMS AVAILABLE FOR TRS-80, APPLE II, PET & ATARI

N.Y.S. residents add sales tax.

All programs require 16K ● TRS-80 programs require LEVEL II BASIC ● APPLE programs require Applesoft BASIC

KRELL SOFTWARE CORP.

"The State of the Art in Educational Computing"

21 Millbrook Drive, Stony Brook, N.Y. 11790 (516) 751-5139

DATA CAPTURE 4.0©

The most advanced and easiest to use telecommunications program for use with the MICROMODEM II™ or the Apple COMMUNICATIONS CARD™.

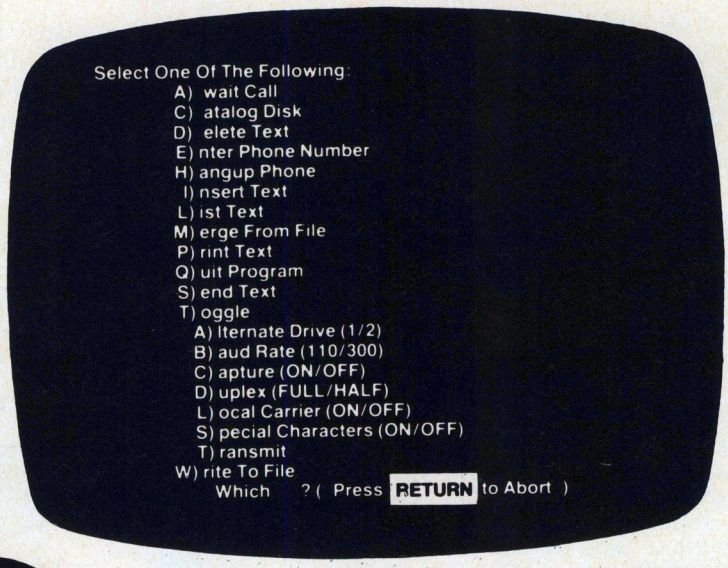
If you use your Apple II™ computer to communicate with other Apples or with timesharing systems, then you need Data Capture 4.0.

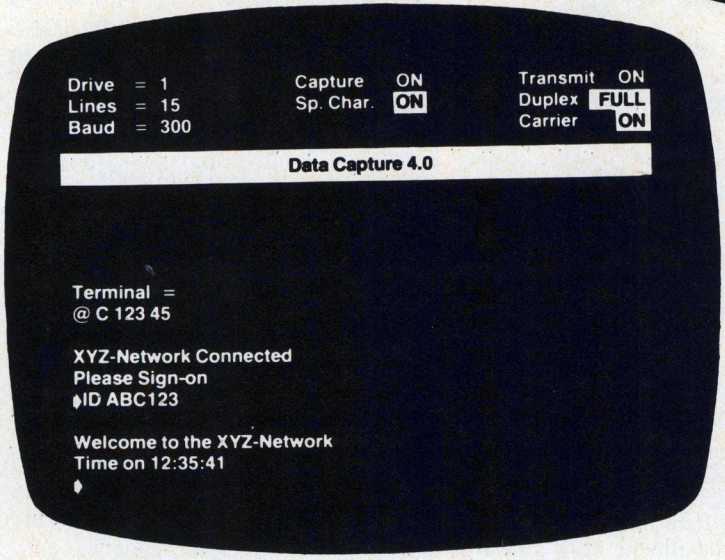
If you want to save stock quotes, airline schedules, electronic mail, or other data from a remote computer system for future use, then you need a smart terminal program for your Apple II™. Data Capture 4.0 is the most user friendly and most flexible program of this type. It is specifically designed to take advantage of all the features of the Micromodem II™, but it works equally well with the Apple Communications Card™ and several other popular serial interface cards.

Data Capture 4.0 is the only Apple II™ smart terminal program available which is fully copyable and modifiable. This means that you don't have to worry about backup. Go ahead and make all the backup copies you need.

The unprotected format and helpful documentation make modifications to Data Capture 4.0 very easy to perform. In addition, full technical support is available by phone from Southeastern Software.

Data Capture 4.0 has many other features. Incoming data files are automatically captured regardless of length. Data in the memory buffer can be viewed, edited, printed, saved to or loaded from disk, or transmitted to the remote system at any time. An unattended mode of operation is provided so that you can call your Apple from another location and send data to it or load data from it. Data Capture 4.0 is fully compatible with the Apple III™ in emulation mode. It is also compatible with all popular lower case adapters for the Apple II™, including the widely used shift key modification. An automatic logon utility for use with the SOURCE is provided with Data Capture 4.0. Also included is a HELP text file containing the latest tips and suggestions on using Data Capture 4.0 effectively (many of these are the result of feedback from customers).





See your local Apple dealer today for a demonstration of Data Capture 4.0 or order direct from Southeastern Software at the address below.

(Please include $2.50 for postage and handling.)

If you presently own a previous version of Data Capture you may upgrade to the new version for the difference in price plus $2.50 for postage and handling. MASTER CARD or VISA orders may be placed by phone.

DATA CAPTURE 4.0 ..... $65.00
DATA CAPTURE 4.0 / 80* $90.00

*Specify either Videx, Smarterm, Double Vision or Sup 'R' Terminal version

— Dealer Inquiries Invited —

Requires DISK II™, Applesoft II™ and 48K of Memory

DATA CAPTURE 4.0© 1980-Southeastern Software

*Apple™, Apple II Plus™, Disk II™ and APPLESOFT II™ are trademarks of Apple Computer Company.

SOURCE – Source Telecomputing Corporation.

*Micromodem™ is a trademark of D.C. Hayes Associates, Inc.

Southeastern Software

6414 Derbyshire Drive ● New Orleans, Louisiana 70126
Dept. CA (504) 246-8438 or (504) 246-7937

master charge THE INTERBANK CARD

BankAmericard VISA

SPECIAL MATH SECTION:

Getting the High Sine

By Charles Kluepfel

I recently completed an Applesoft program which had stored in arrays of character strings 261 formulas of the type such as 2L-3M+180. Each formula was to be evaluated; the result was to be treated as an angle and have its sine or cosine taken, depending on which array it was in. That value would be multiplied by a coefficient from another table and the product added into a total.

The use of a table read in from disk was done for a number of reasons. The overhead of repeating T=T+const*FN SN(....) would have been around 3K above the length of the table, counting line overhead, the remainder of the line besides the formula and coefficient, and the tokens for the *'s that are implicit in the formulas. Combining several on a line was not practical, as that would make checking by sight against the source of the formulas well nigh impossible. Entry would have been more difficult, including extra data besides the formulas, while trying to remember to insert the implied *'s. Also the source, although numbering the formulas and coefficients, did not list the formulas in order, thus necessitating a sort. Having the formulas on a text file also allows for easy updating of any found errors during testing.

This presented the difficulty of having to parse the formulas via a BASIC routine. Needless to say, this interpreter, written in a language that was in turn interpreted, resulted in a slow program. The computation of the elements of an eclipse, which involves invoking a set of 261 formulas four times, took 15 minutes.

I decided to write an assembly language routine to plug any given string (the formulas, one by one) into an existing line of code in BASIC, converting arithmetic symbols into their appropriate tokens, inserting the token for multiplication where implied by a numeric immediately preceding an alphabetic, and also converting variable names such as M' (M prime) to MP for interpretation by BASIC. For good measure it "POKE's" the absolute value of the coefficient of variable M into location 768 (one decimal character converted to its binary value); this feature may be ignored or adapted for use by others.

My program now takes 2 minutes 20 seconds to compute the same eclipse elements that took 15 minutes before.

The machine language routine is invoked with the "&" command, with, for example, & AF$(I) causing the contents of AF$(I), in translated form, to be placed into locations beginning where locations $6 and $7 point. The sample BASIC program shows how these two locations are set up and the ampersand vector at 1013-1015 POKEd in.

```
                    ********************************
                    *                              *
                    *    Ampersand Line Parser     *
                    *     by Charles Kluepfel      *
                    *                              *
                    * Call -A.P.P.L.E. * Mar. 1982 *
                    *                              *
                    ********************************
                    *
                    FILLP    EQU   6
                    SENDP    EQU   8
                    PTRGET   EQU   $DFE3
                    *
                    * Decimal Tokens
                    *
                    RETURN   EQU   177
                    ADD      EQU   200
                    SUBTR    EQU   201
                    MULT     EQU   202
                    DIVD     EQU   203
                    *
                             ORG   $300
                    *
                    COEFF    DS    1
                    Y        DS    1
                    FL       DS    1
                    CT       DS    1
                    *
0304: 20 E3 DF               JSR   PTRGET
0307: 85 08                  STA   SENDP
0309: 84 09                  STY   SENDP+1
030B: A0 00                  LDY   #0
030D: B1 08                  LDA   (SENDP),Y
030F: F0 47                  BEQ   END
0311: 8D 03 03               STA   CT
0314: C8                     INY
0315: B1 08                  LDA   (SENDP),Y
0317: AA                     TAX
```

GETTING THE HIGH SINE

```
0318:  C8          39             INY
0319:  B1 08       40             LDA   (SENDP),Y
031B:  85 09       41             STA   SENDP+1
031D:  86 08       42             STX   SENDP
031F:  A2 00       43             LDX   #0
0321:  8E 02 03    44             STX   FL
0324:  8E 00 03    45             STX   COEFF
0327:  A0 00       46             LDY   #0
0329:  A1 08       47  CONT       LDA   (SENDP,X)
032B:  C9 27       48             CMP   #''
032D:  F0 38       49             BEQ   APOST
032F:  C9 41       50             CMP   #'A
0331:  10 3E       51             BPL   ALPH
0333:  C9 30       52             CMP   #'0
0335:  10 34       53             BPL   NUM
0337:  C9 2B       54             CMP   #'+
0339:  F0 76       55             BEQ   PLUS
033B:  C9 2D       56             CMP   #'-
033D:  F0 76       57             BEQ   MINUS
033F:  C9 2A       58             CMP   #'*
0341:  F0 76       59             BEQ   TIMES
0343:  C9 2F       60             CMP   #'/
0345:  F0 76       61             BEQ   DIVIDE
0347:  8E 02 03    62  PSTO       STX   FL
034A:  91 06       63  STORE      STA   (FILLP),Y
034C:  E6 08       64  INCR       INC   SENDP
034E:  D0 02       65             BNE   OK3
0350:  E6 09       66             INC   SENDP+1
0352:  C8          67  OK3        INY
0353:  CE 03 03    68             DEC   CT
0356:  D0 D1       69             BNE   CONT
0358:  A9 3A       70  END        LDA   #':
035A:  91 06       71             STA   (FILLP),Y
035C:  A9 B1       72             LDA   #RETURN
035E:  C8          73             INY
035F:  91 06       74             STA   (FILLP),Y
0361:  A9 00       75             LDA   #0
0363:  C8          76             INY
0364:  91 06       77             STA   (FILLP),Y
0366:  60          78             RTS
0367:  A9 50       79  APOST      LDA   #'P
0369:  D0 DC       80             BNE   PSTO

036B:  8D 02 03    81  NUM        STA   FL
036E:  4C 4A 03    82             JMP   STORE
0371:  C8          83  ALPH       INY
0372:  91 06       84             STA   (FILLP),Y
0374:  88          85             DEY
0375:  91 06       86             STA   (FILLP),Y
0377:  C9 4D       87             CMP   #'M
0379:  F0 0F       88             BEQ   M
037B:  AD 02 03    89  FLCHK      LDA   FL
037E:  F0 CC       90             BEQ   INCR
0380:  8E 02 03    91             STX   FL
0383:  A9 CA       92             LDA   #MULT
0385:  91 06       93             STA   (FILLP),Y
0387:  C8          94             INY
0388:  D0 C2       95             BNE   INCR
038A:  8C 01 03    96  M          STY   Y
038D:  AD 03 03    97             LDA   CT
0390:  C9 01       98             CMP   #1
0392:  F0 08       99             BEQ   NOTAP
0394:  A0 01      100             LDY   #1
0396:  B1 08      101             LDA   (SENDP),Y
0398:  C9 27      102             CMP   #''
039A:  F0 0F      103             BEQ   RSTY
039C:  AD 02 03   104  NOTAP      LDA   FL
039F:  F0 05      105             BEQ   ONE
03A1:  38         106             SEC
03A2:  E9 30      107             SBC   #$30
03A4:  D0 02      108             BNE   CO
03A6:  A9 01      109  ONE        LDA   #1
03A8:  8D 00 03   110  CO         STA   COEFF
03AB:  AC 01 03   111  RSTY       LDY   Y
03AE:  4C 7B 03   112             JMP   FLCHK
03B1:  A9 C8      113  PLUS       LDA   #ADD
03B3:  D0 92      114             BNE   PSTO
03B5:  A9 C9      115  MINUS      LDA   #SUBTR
03B7:  D0 8E      116             BNE   PSTO
03B9:  A9 CA      117  TIMES      LDA   #MULT
03BB:  D0 8A      118             BNE   PSTO
03BD:  A9 CB      119  DIVIDE     LDA   #DIVD
03BF:  D0 86      120             BNE   PSTO

--END ASSEMBLY--  ERRORS: 0    193 BYTES
```

```
10  REM

    STRING PARSER  DEMO
    by Charles Kluepfel

Call -A.P.P.L.E. Mar. 1982

50 D$ =  CHR$ (4)
60  GOSUB 1000: GOTO 2000
1000 FW =  PEEK (121) + 256 *  PEEK
     (122) + 8: POKE 6,FW - 256 *
      INT (FW / 256): POKE 7, INT
     (FW / 256): POKE 1013,76: POKE
     1014,4: POKE 1015,3: PRINT D
     $"BLOADPARSER": RETURN : REM

Line to be clobbered

2000 AF$(1) = "2L-3M'+180":AF$(2)
     = "5D+M-20":AF$(3) = "3M+2L
     "
2005 L = 35:MP = 22:D = 101:M = 3

2010  FOR I = 1 TO 3
2020  & AF$(I): GOSUB 1000
2030  LIST 1000: PRINT FW, PEEK (
     768)
2040  NEXT I
```

```
]RUN

1000 FW = 2 * L - 3 * MP + 180: RETURN
184               0
1000 FW = 5 * D + M - 20: RETURN
488               1
1000 FW = 3 * M + 2 * L: RETURN
79                3
```

```
]LIST

10 A = 1
20 WIDTH = 33:LEFT = 32
30  HOME : LIST 10
40  POKE WIDTH,20: POKE LEFT,20
50 A = ( PEEK (2055)) + 1: SPEED=
     A
60  VTAB 24: HTAB 36: PRINT  CHR$
     (A)
70  TEXT : IF A < 253 GOTO 30
80  REM

So What Did You Expect?

by Bob Perkins
```

DISCOUNT
DATA PRODUCTS

presents

APPLE SOFTWARE

***** GAMES *** UTILITIES *** BUSINESS *****

Software at Affordable Prices for the budget minded individual

30% OFF
+SPECIAL ITEMS DISCOUNT

APPLE		LIST PRICE	OUR PRICE
★ ADVENTURE INTERNATIONAL			
★ ASTAR INTERNATIONAL COMPANY			
ARTSCI			
Magic Window	D	99.95	79.95
Basic Mailer	D	69.95	55.95
Magic Spell	D	55.95	39.65‡
★ AUTOMATED SIMULATIONS			
★ AVALON HILL GAME COMPANY			
AVANT-GARDE CREATIONS			
Hi-Res Secrets	D	125.95	99.95‡
Hi-Res Golf	D	29.95	23.95
BEAGLE BROTHERS			
DOS Boss	D	24.00	19.95
Utility City	D	29.50	23.95
Alpha Plot	D	39.50	31.95
BRODERBUND SOFTWARE			
Alien Typhoon	D	24.95	19.95
Apple Panic	D	29.95	23.95
Space Quarks	D	29.95	23.95
Genetic Drift	D	29.95	23.95
Red Alert	D	29.95	23.95
David's Midnight Magic	D	34.95	27.95
Arcade Machine	D	44.95	35.95
Track Attack	D	29.95	23.95
BUDGECO			
Raster Blaster	D	29.95	23.95
CALIFORNIA PACIFIC			
Apple-oids	D	29.95	20.65‡
Aklabeth	D	34.95	23.95
CAVALIER COMPUTER			
Aster. Field	D	24.95	19.95
Bug Attack	D	29.95	20.65‡
Star Thief	D	29.95	23.95
★ CONTEXT			
★ CONTINENTAL SOFTWARE			
DAKIN 5/LEVEL 10			
Programming Aids	D	90.00	71.95
Dragon Fire	D	39.95	31.95
Kaves of Karkhan	D	39.95	31.95
Alkemstone	D	39.95	31.95
DATAMOST			
County Fair	D	29.95	20.65‡
Snack Attack	D	29.95	20.65‡
Thief	D	29.95	20.65‡
★ DATASOFT ★ DELTA SOFTWARE			
DENVER SOFTWARE			
Financial Partner	D	175.00	139.95
★ EDU-WARE			
GEBELLI SOFTWARE			
Firebird	D	29.95	23.95
Russki Duck	D	34.95	29.95
HAYDEN SOFTWARE			
Sargon II	D	34.95	27.95
Applesoft Compiler	D	200.00	159.95
★ HIGHLANDS COMPUTER SERVICES			
HOWARD SOFTWARE			
Tax Preparer	D	150.00	119.95
Real Estate Analyzer	D	150.00	119.95

		LIST PRICE	OUR PRICE
INNOVATIVE DESIGN SOFTWARE			
Pool 1.5	D	34.95	27.95
Shuffleboard	D	29.95	23.95
Trick Shot	D	39.95	31.95
INFOCOM			
Zork I	D	39.95	31.95
Zork II	D	39.95	31.95
★ ISA ★ LAZER SYSTEMS ★ IUS			
★ LJK ENTERPRISES ★ MICRO LAB			
★ MICRO PRO INTERNATIONAL			
MICROSOFT			
Typing Tutor	D	19.95	15.95
TASC Compiler	D	175.00	139.95
A.L.D.S.	D	125.00	99.95
★ MICROSOFTWARE SYSTEMS ★ M&R			
★ MUSE			
ON-LINE SYSTEMS			
#2 Wiz & Princess	D	32.95	25.95
#3 Cranston Manor	D	34.95	27.95
#4 Ulysess & Golden Fleece	D	34.95	24.65‡
Superscribe II	D	129.95	103.20
Expeditor II AppleSoft Com.	D	99.95	79.95
Jawbreaker	D	29.95	23.95
Road Work	D	39.95	21.95
Marauder	D	34.95	27.95
Softporn Adv. (X-rated)	D	29.95	23.95
PERSONAL BUSINESS SYSTEM			
Exec. Secretary	D	250.00	195.95
★ PERSONAL SOFTWARE			
PICCADILLY			
Falcons	D	29.95	23.95
Suicide	D	29.95	20.65‡
★ PHOENIX SOFTWARE			
★ QUALITY SOFTWARE			
★ RIVERBANK SOFTWARE			
SENSIBLE SOFTWARE			
Super Disk Copy III	D	30.00	23.95
Multi Disk Cata III	D	25.00	19.95
Disk Recovery	D	30.00	23.95
Applesoft-Plus	D	25.00	19.95
Applesoft Pro. Opt.	D	20.00	15.95
Disk Org. II	D	30.00	23.95
DOS Plus	D	25.00	19.95
Quickloader	D	25.00	19.95
Apple Speller	D	75.00	52.50‡
Image Printer	D	40.00	31.95
SENTIENT SOFTWARE			
Oo-TOPOS	D	32.95	25.95
★ SIERRA SOFTWARE			
SIRIUS SOFTWARE			
Gorgon	D	39.95	31.95
Sneakers	D	29.95	20.95
Epoch	D	34.95	24.50
Cops & Robbers	D	34.95	24.65‡
Outpost	D	29.95	23.95
Beer Run	D	34.95	27.95
Dark Forest	D	29.95	23.95
Borg	D	29.95	23.95
Hadron	D	34.95	27.95
Joyport	A	74.95	59.95

		LIST PRICE	OUR PRICE
SIR-TECH SOFTWARE			
Wizardy	D	49.95	39.95
★ SOFTAPE ★ SOFTWARE EMPORIUM			
SOFTWARE PUBLISHING CORP.			
PFS	D	95.00	66.50‡
PFS Report	D	95.00	71.95
SOUTHEASTERN			
DAta Capture	D	64.95	52.95
Superterm 80 col.	D	90.00	69.95
Smart Term 80 col.	D	90.00	69.95
Videx 80 col.	D	90.00	69.96
SOUTHWESTERN DATA SYSTEMS			
Ascii Express II	D	79.95	63.95
Doubletime Printer	D	149.95	119.95
Financial Mngmnt Sys II	D	59.95	47.95
Listmaster	D	39.95	31.95
Merlin	D	64.95	51.95
The Routine Machine	D	64.95	51.95
Speed Star	D	134.95	107.95
The Correspondent	D	59.95	47.95
Z-Term	D	99.95	79.95
Z-Term "The Pro"	D	150.00	119.95
STONEWARE PRODUCTS			
D.B. Master	D	229.00	160.65‡
D.B. Mas. Utly Pak I	D	99.00	79.95
D.B. Mas. Utly Pak #2	D	99.00	79.95
D.B. Master Stat Pak	D	99.00	79.95
STRATEGIC SIMULATIONS			
Torpedo Fire	D	59.95	47.95
Warp Factor	D	39.95	31.95
Com. Quarterback (2nd Ed.)	D	39.95	31.95
President Elect	D	39.95	31.95
Battle of Shiloh	D	39.95	31.95
Tigers in Snow	D	39.95	31.95
★ SUBLOGIC			
SYNERGISTIC SOFTWARE			
Higher Text II	D	40.00	31.95
Higher Graphics II	D	35.00	27.95
TG PRODUCTS			
Game Paddles	A	39.95	31.95
Joystick	A	59.95	47.95
Expand-a-Port	A	59.95	41.95‡
TURNKEY SOFTWARE			
The Gauntlet	D	34.95	27.95
UNITED SOFTWARE OF AMERICA			
Appleworld	D	59.95	47.95
3-D Supergraphics	D	39.95	31.95
Super KRAM	D	175.00	139.95
Conquest	D	225.00	179.95
VERSA COMPUTING			
VersaWrit Exp. Pac-1	D	39.95	32.95
VersaWrit Gra. Tab	A	299.00	249.95
EZ Port	D	24.95	20.95
★ VIDEX INC. ★ VOYAGER SOFTWARE			

★ SEND FOR OUR PRICE SHEET FOR SPECIFIC PROGRAMS AND PRICES
CALL FOR WEEKLY SPECIALS
PLEASE CHECK FOR SYSTEM REQUIREMENTS BEFORE ORDERING
INCLUDE PUBLISHER AND PRODUCT NAME
FOREIGN INQUIRIES INVITED — ADD 10% SHIPPING

All Orders Prepaid by Person Check, Money Order, Cashier check
For Faster Delivery Srvice C.O.D. — Cash Only/Master Charge — Visa
Mastercharge — Visa add 3%, and include expiration date on card
Please Add $2.50 for Shipping

California Residents Add 6% Sales Tax
Prices Valid Through This Month's Issue
Prices Subject to Change Without Notice
Please Specify Tape or Disk

DIRECT ORDER INQUIRIES TO:
DISCOUNT DATA PRODUCTS
P.O. BOX 19674-AC
SAN DIEGO, CA 92119
PHONE 714-287-0190

Professional proofreading on the APPLE II Computer is now a reality

THE APPLE SPELLER fills the void that has consistently kept the large variety of excellent word processing packages for the Apple II Computer from approaching the power of a dedicated work processor. Finally, the first professional quality spelling verification program is available for the Apple II. The Apple Speller will certainly be the standard against which all other similar programs are compared.

The Apple Speller interfaces to the most popular Apple word processors, including Applewriter, Apple Pie, Executive Secretary, Letter Perfect, Magic Window, and Superscribe II, just to name a few. In fact, The Apple Speller can analyze the output of any editor that writes a standard Apple binary or text file to a diskette. In addition to this flexibility, the performance of The Apple Speller will astound the microcomputer world.

The Apple Speller is supplied with a 30,000+ word dictionary on a single 5¼" diskette with additional space to easily add another 8,000 words to suit your individual needs.

The Apple Speller has built-in utilities to maintain the dictionary diskette. You can readily add words, delete words, and create an unlimited number of modified and/or new dictionaries for specific application.

The Apple Speller is unbelievably fast. The first pass reads your document and collects all the words it contains at a rate of 5,000 words per minute. Next, the words are compared to the dictionary for spelling errors at the incredible speed of 50,000 words per minute. Finally, all misspelled words are marked as such in your document with a rate of 1,000 words per minute. This translates to proofreading a 10 page document in 1 minute if there are no spelling mistakes and 2 minutes, 15 seconds for an unlimited number of spelling errors.

Numerous options are provided throughout the program to enable you to completely control all activities of The Apple Speller. These include the ability to ignore both control codes and formatting commands, an alphabetical listing of either mispelled words or all the words in your document along with usage frequencies, multiple options for the action taken with each misspelled word, and much, much more! A verification mode is provided to allow you to examine and dispense with misspelled words while viewing them in the actual context in which they appeared in your file.

The Apple Speller requires an Apple II/Apple II+ equipped with 48K, DOS 3.3, and 1 or 2 disk drives. Two disk drives are required to delete or add words to the dictionary.

The Apple Speller is being introduced at the incredible price of **$75.00!**

Sensible Software
6619 Perham Drive Dept. C
West Bloomfield, Michigan 48033
(313) 399-8877
Visa and Mastercard Welcome
Please add $1.25 postage and handling per diskette
Please specify program desired

APPLE is a registered trademark of APPLE Computer Company Registered Trademark of DIGITAL RESEARCH CORP. Registered Trademark of MUSE SOFTWARE

TRIG OR TREAT

By Raymond Usher

APPLESOFT Basic has the three most used trig functions built in, and the other three are the reciprocals of those; so calculating the function of an angle is simple if the angle is expressed in radians.

However, Applesoft lacks five of the six possible inverse trig functions. These functions are derived using fairly long equations. (See page 103 of the Applesoft Reference Manual). Besides being long, the use of a constant in four of the five expressions is unnecessary since each function is intrinsically related to all the other functions. (i.e. If you are given one, you can find all the others).

One more problem is created by using a constant; accuracy is limited to the number of digits used in the constant, in this case four. The constant represents the number of radians in 90 degrees and could be lengthened to 1.5707963-27, but that would make the expression longer and . . . there is a better way.

The following are the five functions in their relational form using the built-in function, ATN.

```
ARCSIN(X)  = ATN(1/SQR((1/X)
             ∧2-1))
ARCCOS(X)  = ATN(SQR((1/X)
             ∧2-1))
ARCSEC(X)  = ATN(SQR(X∧2-1))
ARCCSC(X)  = ATN(1/SQR
             (X∧2-1))
ARCCOT(X)  = ATN(1/X)
```

A word of caution. The Apple doesn't handle angles larger than 90 degrees (PI/2 radians). However, subtracting 90 degrees (PI/2 rad) from a larger angle until you're dealing with an angle of less than or equal to 90 (PI/2), and changing the sign of the function of that angle for each 90 degrees (PI/2 rad) subtracted will yield the correct result.

Finding the angle from a negative function of an angle can be done by changing the sign, using the correct inverse function, and adding 90 degrees (PI/2 rad) to the answer.

The short program that accompanies this article demonstrates the validity of the relational function, shows the decreased accuracy of the inverse function found in the manual, and impliments an easy conversion to degrees. (All the Apple trig functions use radian measure).

```
]LIST

5  REM

TRIG DEMO by Raymond S Usher

Call -A.P.P.L.E. * Mar. 1982

10 PI = 3.14159265:C = 360 / (2 *
     PI): REM DEG TO RAD CONSTANT

20  INPUT "INPUT ANGLE IN DEGREES
    ...";N
30 X =  COS (N / C)
40  PRINT X;"... COS OF N"
45  PRINT : PRINT " USING APPLES
    FN"
50 Z =  -  ATN (X /  SQR ( - X *
    X + 1)) + 1.5708
60  PRINT Z;"...ANGLE IN RAD"
70  PRINT Z * C;"...ANGLE IN DEG"

80 Y =  ATN ( SQR ((1 / X) ^ 2 -
    1))
85  PRINT : PRINT " USING RELATIO
    NAL FN"
90  PRINT Y;"...ANGLE IN RAD"
100  PRINT Y * C;"...ANGLE IN DEG
    "
120  END


]RUN
INPUT ANGLE IN DEGREES ...60
.500000001... COS OF N

 USING APPLES FN
1.04720122...ANGLE IN RAD
60.0002105...ANGLE IN DEG

 USING RELATIONAL FN
1.04719755...ANGLE IN RAD
60...ANGLE IN DEG
```


For A.P.P.L.E. Members Only

SYMBOL SYMON

MEMBER PRICE
$20.00

SYMBOL SYMON is a monitor and a symbolic debugger for the Apple][/][+ system. It combines step/trace functions with breakpoint tools, register and status flag modification commands, easily discernible status flag displays, ASCII memory dumps and user dedicated stack and zero page. Furthermore, if the DOS Tool Kit, Big Mac, or Ted][+ is used for assembly, hex addresses can be represented very conveniently by their symbolic equivalents.

SYMBOL SYMON is not an isolated system. The DOS and Apple monitor control facilities are still at the disposal of the user.

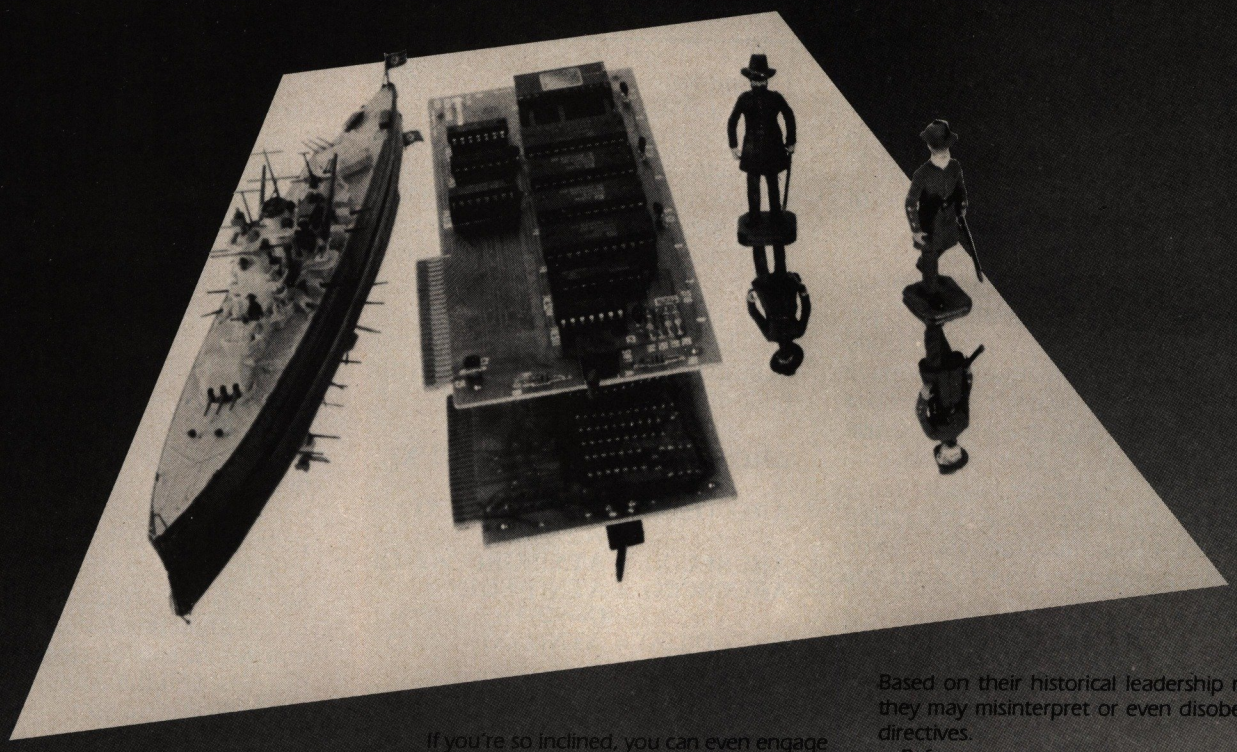
A.P.P.L.E. Orders
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

Overseas Shipping Wt: 6 oz.

Washington state residents add 6.6% sales tax.

All foreign orders (incl. Canada) will require additional postage and shipping.

WE BRING THE FUTURE TO THE PAST.



The future belongs to microcomputers.

At SSI, we apply tomorrow's programming techniques to re-create some classic battles of the past. As proof, we offer our latest games for the Apple® computer*: PURSUIT OF THE GRAF SPEE and THE ROAD TO GETTYSBURG.

PURSUIT OF THE GRAF SPEE™ transports you back to the South Atlantic of 1939 so you can command the deadly German pocket battleship to sink enemy merchant vessels while eluding British and French warships. Or you can direct the Allied fleet to locate and destroy the scourge of your shipping lanes.

If you're familiar with **Computer Bismarck** — which has become something of a phenomenon in gaming circles — you owe it to yourself to try this game. Like its older brother, PURSUIT OF THE GRAF SPEE has all the thrill of the chase, heightened by the complexity and richness of detail such as complete ship ratings for speed and strength, fuel restrictions, refueling capabilities, and realistic rules governing sighting and visibility.

However, we've also added a remarkable tactical combat system that accounts for gunfire accuracy, torpedoes and smoke screens. When it comes to calculating ship damage and battle results, you simply leave all the work to the computer, which does everything instantly and unerringly.

If you're so inclined, you can even engage in combat by using miniatures. Enter the results of your off-line battle and the computer will incorporate them into the strategic game.

This simulation contains two scenarios, all played on a 19-by-19 square-grid map of the South Atlantic displayed in colorful Hi-Res graphics. The first recounts the entire 25-day historic chase, where shadowing, pursuit and elusive maneuvers are your primary concerns. The second simulates the Battle of the River Plate, where the Graf Spee — already located — must deal with three British ships.

Finally, the solitaire scenario lets you match wits against the computer as it directs the Graf Spee. We wish you luck.

THE ROAD TO GETTYSBURG™ takes you even further back in time, to that fateful battle of 1863. This board-assisted computer simulation is designed to let you experience the actual feel of a Civil War command. Like a real Union or Confederate general, you must deal with the slow and frustrating dispatch system — your only means of communication with your troops. Knowledge of troop positions and estimates of enemy strength are only as good as your reconnaissance patrols.

The computer plays the individual corps commanders who implement your orders. Based on their historical leadership ratings, they may misinterpret or even disobey your directives.

Before you move a corps, you must consider the effects of weather, terrain, corps morale, fatigue, and even straggling on your troops.

THE ROAD TO GETTYSBURG simulates the entire week-long campaign. Depending on your strategies, the town of Gettysburg may not be involved at all.

For those who want to dispense with long-range planning, we've provided a scenario that just simulates the three-day Battle of Gettysburg. There is also a solitaire scenario so you can play anytime against a tough computer opponent.

Again, you can use miniatures to resolve all battles if you so desire.

This game is made for all you avid strategists out there. After you're done playing it, you'll really know the pleasure and pain of being a Robert E. Lee!

If you're interested in what the future has in store for the past, visit your local computer/game store and get these SSI games today. At $59.95 each, they must surely be the best bargain in time travel!

To order with your VISA or MC, call **toll free 800-227-1617, ext. 335** (800-772-3545, ext. 335 in California). To order by mail, send your check to Strategic Simulations Inc, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043. All our games carry a 14-day money back guarantee.

*48K Apple® II with ROM and one floppy disc drive.

SSI

Apple is a registered trademark of Apple Computer Inc.

STRATEGIC SIMULATIONS INC

As part of our demanding standards of excellence, we use maxell floppy discs.

WE'RE LOOKING FOR A FEW GOOD DESIGNERS. **We're always on the lookout for high-quality games from independent designers. If you have game programs that you'd like SSI to publish, give us a call at (415) 964-1353 or write to SSI, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043.**

APPLEMASH:

MATHEMAGICS

By Michael Thyng

AND now, for you faithful fans of the mashed Apple, I bring you yet another capability for your Apple. Today we're going to add extremely large numbers without losing the precision of scientific notation.

Let's see why there would be a need for a routine to add very large numbers. Suppose you want to store a numeric variable in memory with a value of 999,999.99. You could write 100 A = 999999.99. When you went to print that out it would print 999999.99. Now, suppose you store another variable B. Let us enter 200 B = 9999999.99. Notice that B has 7 nines to the left of the decimal point. Now print B out. You will get 10000000. Let's try C. At line 300 set C = 123456789.21. Now print it out. (see exhibit 1) At line 400 set D = 12345678.19. Now print it out. And finally, at line 500, set E = 1234567892233.

What do you notice? Well, the common thread of similarity is that the outputs all have a precision less than or equal to nine digits. (again, see exhibit 1).

In some cases this isn't even a concern. In games, for example, a calculation that yielded a number as large as 10 million would be just as good if it was within a few hundred or even a few thousand. But for the business concerns that we regularly talk about, it just won't do. Now, for decision making, based on fairly accurate estimates. OK, sure, it would be fine. But for bookkeeping type calculations, roundings of this sort are not acceptable. So, we're going to develop an algorithm that will allow us to add numbers as large as 99 trillion.

Where shall we begin? Let's start with what we know. We know that we can add numbers that have fewer than nine digits and maintain our precision. We know that the Apple makes it easy for us to call subroutines. We know that we can easily convert from numeric mode to alphanumeric mode and back again. We know that a string variable can have 255 characters. So let's put something together that will solve our problem.

To begin, let's choose to work with all very large number calculations in strings. That way we always maintain the digits we started with. Now let us choose some very large numbers as our test case examples. Also, let's work in the context of an Applesoft program for clarity and commonality of understanding. Fine. First, a REM statement to label our work. (refer to exhibit 2). Next, we will choose two variables and give them the values we want to add. I chose A$ and B$ for us. You're welcome. I kind of thought you'd like them. A$ now equals 111,222,333, 666,555.77 and B$ equals 444,555, 666,555,111.33. Exhibit 2 shows a GOTO 100 at line 39.

Logic dictates that the next thing we want to do is set up to add. To do that we have to break our number up into pieces that can be added without losing the precision that we are trying to maintain. So, what is logical? Right. We will break our numbers up into nine digit increments. Now, at this point I made a decision. Just how big is big? I mean, even though I am writing a routine to handle huge numbers there needs to be a limit, right? Well, I said *right* when I answered that question. And the limit I chose was 999 trillion. I chose that limit because I don't know what the name of the next higher thousand is. Besides even the government hasn't elected to go into debt that high (yet).

Look again at A$. It has 17 digits of precision and represents over 100 trillion. So, all we need to do is break it up into two numeric pieces. I let AL represent the lower half of A$ and AH represent the value of the higher half. Now, those of you who may not have done so before, but like esoteric programming, could modify the routine we have laid out here to handle much larger numbers. At line 100 we are taking the lower nine characters of A$ with the RIGHT$ command and converting them to a numeric value with the VAL command.

At line 110 we are taking the upper characters and converting them to the numeric equivalent of the higher portion of A$. Line 110 appears a bit more complicated than line 100 because we don't know just how big A$ will be as we use it in a program. In line 110, we want to assign variable AH the numeric value of the left portion of A$. To allow for a variety of sizes of A$, we say we will take the LEFT$ portion of A$ of length LEN (A$) minus 9. Remember, we already took nine characters out of A$. We neither want to duplicate any of the digits or delete any. So, line 110 takes what's left.

Suppose A$ has a length of less than nine characters. Lines 105 and 205 take care of that problem. Lines 200 and 210 handle B$'s high and low values. Line 300 does the low end addition of AL and BL. Then we have to consider another problem. Just because AL and BL are in the precision range that works so well on the Apple, we can't forget that CL, the sum, may have exceeded the precision range. I introduced a new variable, CX, to handle the overflow of the low order addition. That overflow value from the low portion must be added in to the high order values AH and BH. Now we could construct sum variable C$, and we do so at line 400.

The next thing we need to do is print out the large number contained in C$. But to improve readability we would format it. The formatter is stored at lines 40 through 80 and needs just a little explanation. As before, we don't know just how big a number we're dealing with. And the length of the numbers we send through this subroutine will vary from one large number to the next. So the routine counts in line 50 from the right hand side of the number to the highest digit on the left. Along the way it puts commas where appropriate. Lines 510 to 530 revalue A$ and B$ after they've gone through the Formatter subroutine. Lines 602 to 650 print out the addition in a standard format that lines up the decimal points. Actually lines 602,604, and 606 do that.

Now that we've done it once, we should generalize the routines so they can be used in other programs. First we'd delete lines 20 through 39. Then we would have to decide how many times we would be accessing this routine. It isn't a very speedy process to convert back and forth between alphanumeric and numeric variables and vice-versa. So we want to minimize the use. Finally we have to decide whether the formatting of lines 602 through 650 would work for us or whether we have to have another kind of format. See "Amplifying Applesoft," *Call -A.P.P.L.E. in Depth no. 1,* for a couple of additional formatting routines.

This routine shows how the Apple can do the same things as some of the "number crunching" machines. And also demonstrates one of the differences between the two-speed. But it isn't a problem for me. If it takes too long to run a stream of calculations, I could always run it at night and sleep while my faithful Apple crunches away.

```
]LIST

5  REM

 APPLEMASH  by Mike Thyng

Call -A.P.P.L.E. Mar. 1982

7  REM EXHIBIT 1

100 A = 999999.99
200 B = 9999999.99
300 C = 123456789.21
400 D = 12345678.19
500 E = 1234567892233
910   PRINT "A = ";A
920   PRINT "B = ";B
930   PRINT "C = ";C
940   PRINT "D = ";D
950   PRINT "E = ";E


]RUN
A = 999999.99
B = 10000000
C = 123456789
D = 12345678.2
E = 1.23456789E+12
```

```
5  REM

 APPLEMASH  by Mike Thyng

Call -A.P.P.L.E. Mar. 1982

7  REM EXHIBIT 2

10   REM LARGE NUMBER ADDER

20 A$ = "111222333666555.77"
30 B$ = "444555666555111.33"
39   GOTO 100
40   REM $ FORMATTER

45 W$ = "":A = 0
50   FOR N =  LEN (U$) TO 1 STEP  -
       1
55 W$ =  MID$ (U$,N,1) + W$
60 A = A + 1
65   IF  MID$ (U$,N,1) = "." THEN
       A = 0
67   IF N = 1 THEN 80
70   IF A = 3 THEN W$ = "," + W$:A
       = 0
75   NEXT N
80   IF  LEFT$ ( RIGHT$ (W$,2),1) =
       "." THEN W$ = W$ + "0"
90   RETURN
100 AL =  VAL ( RIGHT$ (A$,9))
105   IF  LEN (A$) < 9 THEN 200
110 AH =  VAL ( LEFT$ (A$,( LEN (
       A$) - 9)))
200 BL =  VAL ( RIGHT$ (B$,9))
205   IF  LEN (B$) < 9 THEN 300
210 BH =  VAL ( LEFT$ (B$,( LEN (
       B$) - 9)))
300 CL = AL + BL
320   IF CL <  = 999999.99 THEN 39
       0
340 CX =  INT (CL / 1000000)
360 CL = CL - CX * 1000000
390 CH = AH + BH + CX
400 C$ =  STR$ (CH) +  STR$ (CL)
510 U$ = A$: GOSUB 40:A$ = W$
520 U$ = B$: GOSUB 40:B$ = W$
530 U$ = C$: GOSUB 40:C$ = W$
602   IF  LEN (A$) < 25 THEN A$ =
       " " + A$: GOTO 602
604   IF  LEN (B$) < 25 THEN B$ =
       " " + B$: GOTO 604
606   IF  LEN (C$) < 25 THEN C$ =
       " " + C$: GOTO 606
610   PRINT "  ";A$
620   PRINT " +";B$
630   PRINT "     ----------------
       ------"
650   PRINT "  ";C$

]RUN
      111,222,333,666,555.77
 +    444,555,666,555,111.33
     ----------------------
      555,778,000,221,667.10

]125 A$ = "111.44"

]RUN
                      111.44
 +    444,555,666,555,111.33
     ----------------------
      444,555,666,555,222.77
```

finally . . .

LOW COST, QUALITY STRATEGIC SOFTWARE

WE OPERATE ON HIGH VOLUME — NOT HIGH PRICES

*** JOB SCHEDULER** $19.95
- Uses Critical Path Algorithm
- Business, Project or Multiple Home Tasks

*** FINANCIAL EVALUATOR** $19.95
- Investment Returns (IRR & Present Value)
- Up to 95 Different Cash Flows
- Easy Screen Entry

*** BUSINESS ANALYZER** $21.95
- After Tax Returns on Business Investments

(For Apple II® + with 3.3 DOS)
Apple is a Trademark of Apple Computer, Inc.
Send Check or Money Order To:

STRATEGIC SOFTWARE

1259 El Camino Real, Suite 600
Menlo Park, CA 94025

- Add $2.00 for postage and handling
- California residents add 6% sales tax

COMPUTER CONSULTING SERVICES

TRUST ACCOUNT MANAGEMENT SYSTEM

1376 CLEVELAND STREET
CLEARWATER, FLA. 33515
(813) 441-3490
(813) 446-2643

The people that brought you the Professional Timekeeping System are proud to announce the introduction of another valuable, time-saving office management tool . . . the C.C.S. Trust Account Management System. Designed with consultation from bankers and legal office managers, this System incorporates all of the principles of basic trust account management into a simple, yet powerful office accounting tool. It expands on these basic principles to give the user instant access to often-requested data such as client trust balances and audit trails for individual client accounts or entire trust accounts.

As an additional feature, C.C.S. has provided a function in this system which will automatically transfer funds from any Trust System account into a client account in the Professional Timekeeping System, making this System a perfect complement to that already powerful legal office management system.

As is our tradition, this system was designed for professionals by professionals. See your local Apple dealer for details.

Hardware Requirements: APPLE II PLUS (48k), DOS 3.3
Dual disk drives, printer

Dealer Inquiries Invited

System Functions:
- Open a new trust account
- Enter deposits and withdrawals
- Print withdrawal checks
- Print account transactions
- Print check/deposit registers
- Void entries
- Analyze an entire trust account
- Analyze a client's account
- Reconcile monthly bank statement
- Purge an existing account
- Directory assistance
- Edit account information
- Change System parameters
- Change System password

Capacities:

One trust account per disk. The System will accommodate any number of accounts. Specific capacities are dependent on the number of client accounts within the trust account.

Typical configuration:
200 clients
1600 accounts entries
8 entries per client, average

(These figures are for active account entries. Records from client accounts with zero balances are periodically removed with the Purge function.)

Apple II

Apple II

Quality Disk Software from SPECTRUM

PERSONAL FINANCE MASTER:

The premier personal and small business financial system. Covering all types of accounts including check registers, savings, money market, loan, credit card and other asset or liability accounts, the system has these features:

- Monthly Transaction Reports
- Budgets Income & Expense
- Reconciles to Bank Statements
- Prints Checks & Mailing Labels
- Automatic Year-End Rollover
- Prepares a Net Worth Report
- Searches for Transactions
- Handles Split Transactions
- User-Friendly Data Entry Forms
- Fast Machine Language Routines
- Extensive Error Trapping
- HI-RES Expense/Income Plots

For Apple II (48K) $74.95

COLOR CALENDAR: $29.95

Got a busy calendar? Organize it with Color Calendar. Whether it's birthdays, appointments, business meetings or a regular office schedule, this program is the perfect way to schedule your activities. The calendar display is a beautiful **HI-RES** color graphics calendar of the selected month with each scheduled day highlighted in color. Using the daily schedule, you can review any day of the month and schedule an event or activity in any one of 20 time slots from 8:00 A.M. to 5:30 P.M.

BUSINESS SOFTWARE:

Both Programs $249.95

A user-friendly yet comprehensive double-entry accounting system employing screen-oriented data input forms, extensive error-trapping, data validation and machine language routines for high speed operation. The series includes these two modules:

GENERAL LEDGER: A complete accounting system with these features:

- Up to 500 accounts and 500 transactions per month.
- Extensive check register management system (similar to our PFM).
- Prints checks and mailing labels.
- Produces these reports:
  - Transactions Journal
  - Account Ledgers
  - Income Statement
  - Balance Sheet
  - Account Listings

Requires Apple II, 48K RAM, 1 disk drive. $149.95

ACCOUNTS RECEIVABLE: A flexible system with these features:

- Up to 500 accounts and up to 500 invoices per diskette.
- Prints invoices and customer statements & address labels.
- Interfaces to General Ledger.
- Interactive screen-based invoice work sheet.
- Produces these reports:
  - Aged Receivables
  - Sales Analysis
  - Customer Listings
  - Invoice Search

Requires Apple II, 48K RAM, 2 disk drives $149.95

Both Programs $249.95

UNIVERSAL BUSINESS MACHINE:

An electronic spread sheet structured around a 100 row × 20 column table. User defines row and column names and equations forming a unique template. Table elements can be multiplied, added, subtracted, divided, summed, averaged and accumulated. Hundreds of unique templates can be created, used, stored and recalled for later use. Supplied with 8 standard templates ready for use covering these subjects.

- Cash Flow Analysis
- Proforma Profit & Loss
- Proforma Balance Sheet
- Real Estate Investment
- Sales Forecaster
- Source and Use of Funds
- Job Cost Estimator
- Inventory Analysis

Price (Apple II, 48K) $89.95

ELECTRONICS SERIES VOL I & II:

Entire Series $259.95

LOGIC SIMULATOR: SAVE TIME AND MONEY. Simulate your digital logic circuits before you build them. **CMOS, TTL,** or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including **NANDS, NORS, INVERTERS, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS** up to 20 user-defined random, or binary input patterns. Accepts network descriptions from keyboard or from **LOGIC DESIGNER** for simulation (48K) $159.95

LOGIC DESIGNER: interactive **HI-RES** graphics program for designing digital logic systems. Draw directly on the screen up to 10 different gate types, including **NAND, NOR, INVERTER, EX-OR, T-FLOP, JK-FLOP, D-FLOP, RS-FLOP, 4 BIT COUNTER** and **N-BIT SHIFT REGISTER.** User interconnects gates using line graphics commands. Network descriptions for **LOGIC SIMULATOR** generated simultaneously with the **CRT** diagram being drawn $159.95

MANUAL AND DEMO DISK: Instruction Manual and demo disk illustrating capabilities of both programs) $29.95

ELECTRONIC SERIES VOL III & IV:

Entire Series $259.95

CIRCUIT SIMULATOR: Tired of trial & error circuit design? Simulate & debug your designs before you build them! With **CIRCUIT SIMULATOR** you build a model of your circuit using **RESISTORS, CAPACITORS, INDUCTORS, TRANSISTORS, DIODES, VOLTAGE** and **CURRENT SOURCES** and simulate the waveform response to inputs such as **PULSES, SINUSOIDS, SAWTOOTHS,** etc... all fully programmable. The output is displayed as an **OSCILLOSCOPE-STYLE PLOT** of the selected waveforms or as a printed table of voltage vs time. Handles up to 50 nodes and 100 components. Requires 48 **RAM** $159.95

CIRCUIT DESIGNER: Interactive **HI-RES** graphics program for designing electronic circuits. Draw directly on the screen up to 10 different component types, including those referenced above. Component interconnect list for **CIRCUIT SIMULATOR** generated automatically, Requires 48K **RAM** $159.95

MATHEMATICS SERIES:

Entire Series $49.95

STATISTICAL ANALYSIS I: This menu driven program performs **LINEAR REGRESSION** analysis, determines the mean, standard deviation and plots the frequency distribution of user-supplied data sets. Printer, Disk, I/O routines.

..... $19.95

NUMERICAL ANALYSIS: HI-RES 2-Dimensional plot of any function. Automatic scaling. At your option, the program will plot the function, plot the **INTEGRAL,** plot the **DERIVATIVE,** determine the **ROOTS, MAXIMA, MINIMA, INTEGRAL VALUE** $19.95

MATRIX: A general purpose, menu driven program for determining the **INVERSE** and **DETERMINANT** of any matrix, as well as the **SOLUTION** to any set of **SIMULTANEOUS LINEAR EQUATIONS** $19.95

3-D SUFAACE PLOTTER: Explore the **ELEGANCE** and **BEAUTY** of **MATHEMATICS** by creating **HI-RES PLOTS** of 3-dimensional surfaces from any 3-variable equation. Disk save and recall routines for plots. Menu driven to vary surface parameters. Hidden line or transparent plotting $19.95

ACTION ADVENTURE GAMES:

Entire Series $29.95

RED BARON: Can you outfly the **RED BARON?** This fast action game simulates a machine-gun **DOGFIGHT** between your **WORLD WAR I BI-PLANE** and the baron's. You can **LOOP, DIVE, BANK** or **CLIMB**-and so can the **BARON.** In **HI-RES** graphics plus sound $14.95

BATTLE OF MIDWAY: You are in command of the **U.S.S. HORNETS' DIVE-BOMBER** squadron. Your targets are the Aircraft carriers, Akagi, Soryu and Kaga. You must fly your way through **ZEROS** and **AA FIRE** to make your **DIVE-BOMB** run. In **HI-RES** graphics plus sound..... $14.95

SUB ATTACK: It's April 1943. The enemy convoy is headed for the **CONTROL SEA.** Your sub, the **MORAY,** has just sighted the **CARRIERS** and **BATTLESHIPS'.** Easy pickings. But watch out for the **DESTROYERS** - they're fast and deadly. In **HI-RES** graphics plus sound. $14.95

FREE CATALOG-All programs are supplied on disk and run on Apple II w/Disk & Applesoft **ROM** Card and require 32K **RAM** unless otherwise noted. Detailed instructions included. Orders shipped within 5 days. Card users include card number. Add $2.00 postage and handling with each order. California residents add 6½% sales tax. Foreign orders add $5.00 postage and handling per product.

SPECTRUM SOFTWARE
142 Carlow, P.O. Box 2084
Sunnyvale, CA 94087

master charge
VISA

FOR PHONE ORDERS:
(408) 738-4387
DEALER INQUIRIES INVITED.

MOD ART

By Wayne Eastwood

THE published MOD algorithm for Applesoft Basic (Applesoft][Basic Programming Manual, p. 104) is incorrect. The algorithm (Formula 1) works only if both A and B are positive.

Formula 1: A MOD B = INT ((A / B - INT (A / B)) * B + .05) * SGN (A / B)

The absolute value is correct but the sign is wrong if both A and B are negative. This is caused by the multiplication by B, which skews the result if negative. The algorithm fails whenever A and B are opposite in sign. The problem is in the INT function itself. We are prone to treat it as a truncating function (i.e., INT (-2.5) = -2). This is *not* the case, as the BASIC manual explains. It returns the largest integer *less than* or equal to the argument. For positive values this works out fine. But for negative values, the results are not always what one might expect. INT (-2.5) =-3, *not* -2. Remember, the largest integer less than or equal to... Thus when A and B are opposite in sign, two errors occur. The enclosed INT function, the one multiplied by B, is wrong by a factor of B. Additionally, the result computed within the larger INT is shifted low by one, whenever the whole computation is negative (i.e., whenever A is positive and B is negative.) Program 1 with Listing 1, along with Program 2 with Listing 2, show the results of the Integer Basic MOD function, compared with the published Applesoft equivalent.

Pretty bad.

So how do we solve the problem? Well, how does a programmer normally sit down and solve a problem? Pick a short, easy to compute example, and proceed. So let's pick 5 and 2. We need the remainder when 5 is divided by 2. So we derive Formula 2:

Formula 2: X = A - INT (A / B) * B
or X = 5 - INT (5 / 2) * 2

INT (5 / 2) = 2, which when multiplied by 2, equals 4. 5-4 = 1 and we're home. Well, not quite. What if we had chosen -5 and -2? INT (-5 / -2) = 2; so far, so good. Look out, 2 * -2 = -4, and -5 - (-4) = -1. So we devise Formula 3 to work for both cases:

Formula 3: X = (ABS(A) - INT(A/B)*ABS(B))

Now we've taken care of all cases when A and B have the same sign. Already we can see the solution for cases when A and B have opposite signs. We simply put an ABS function into the inner INT function to take care of negative rounding. Convention requires that the MOD of a negative division be negative. So we need to add still another fix to change the sign. That gives us formula 4:

Formula 4: X = (ABS (A) - INT (ABS (A / B)) * AB S(B)) * SGN (A / B)

All that remains is to take care of any possible rounding errors in Applesoft, and insure that the result is an integer, which brings us to formula 5. Program 3 with listing 3 shows the results when using the new MOD algorithm.

Formula 5: INT (ABS (A) - INT (ABS (A / B)) * ABS (B) + .05) * SGN (A / B)

That's it.

LISTING 1
INTEGER MOD FUNCTION

```
]LIST
  100 DIM A$(4)
  110 PRINT "A (ROWS) VARIES FROM -999
      TO 999 IN INCREMENTS OF 127"

  120 PRINT "B (COLUMNS) VARIES FROM -
      201 TO 210 IN INCREMENTS OF 31"

  130 PRINT : PRINT : PRINT "        "
      ;
  140 FOR B=-201 TO 210 STEP 31
  150 A$="    ": IF B)9 OR B(0 THEN
      A$="   ": IF B)99 OR B(-9 THEN
      A$="  ": IF B)999 OR B(-99 THEN
      A$=""
  160 PRINT A$;B;"  ";
  170 NEXT B: PRINT
  180 FOR A=1 TO 100: PRINT "-";:
      NEXT A: PRINT
  190 FOR A=-999 TO 999 STEP 127
  200 A$="    ": IF A)9 OR A(0 THEN
      A$="   ": IF A)99 OR A(-9 THEN
      A$="  ": IF A)999 OR A(-99 THEN
      A$=""
  210 PRINT A$;A;" : ";
  220 FOR B=-201 TO 210 STEP 31
  230 C=A MOD B
  240 REM
  250 REM  PRINT FORMAT
  260 REM
  270 A$="    "
  280 IF C)9 OR C(0 THEN A$="   "
  290 IF C)99 OR C(-9 THEN A$="  "

  300 IF C)999 OR C(-99 THEN A$=""

  310 PRINT A$;C;"  ";
  320 NEXT B: PRINT : NEXT A
  330 END
```

LISTING 2
LINE 230 USING APPLESOFT
MOD FUNCTION AS DESCRIBED
IN APPLESOFT MANUAL

```
230 C =  INT ((A / B -  INT (A /
    B)) * B + .05) *  SGN (A / B
    )
```

LISTING 3
APPLESOFT MOD FUNCTION
as modified by author

```
]LIST

110   PRINT "A (ROWS) VARIES FROM
      -999 TO 999 IN INCREMENTS OF
       127"
120   PRINT "B (COLUMNS) VARIES FR
      OM -201 TO 210 IN INCREMENTS
       OF 31"
130   PRINT : PRINT : PRINT "
       ";
140   FOR B =  - 201 TO 210 STEP 3
      1
150 A$ = "    ": IF B > 9 OR B < 0
       THEN A$ = "  "
151   IF B > 99 OR B <  - 9 THEN A
      $ = " "
152   IF B > 999 OR B <  - 99 THEN
      A$ = ""
160   PRINT A$;B;"  ";
170   NEXT B: PRINT
180   FOR A = 1 TO 90: PRINT "-";:
       NEXT A: PRINT
190   FOR A =  - 999 TO 999 STEP 1
      27
200 A$ = "    ": IF A > 9 OR A < 0
       THEN A$ = "  "
201   IF A > 99 OR A <  - 9 THEN A
      $ = " "
202   IF A > 999 OR A <  - 99 THEN
      A$ = ""
210   PRINT A$;A;" : ";
220   FOR B =  - 201 TO 210 STEP 3
      1
230 C =  INT ( ABS (A) -  INT ( ABS
      (A / B)) *  ABS (B) + .05) *
       SGN (A / B)
240   REM
250   REM  PRINT FORMAT
260   REM
270 A$ = "    "
280   IF C > 9 OR C < 0 THEN A$ =
      "  "
290   IF C > 99 OR C <  - 9 THEN A
      $ = " "
300   IF C > 999 OR C <  - 99 THEN
      A$ = ""
310   PRINT A$;C;"  ";
320   NEXT B: PRINT : NEXT A
330   END
```


COMPUTER CONSULTING SERVICES

IT'S ABOUT TIME . .

1376 CLEVELAND STREET
CLEARWATER, FLA. 33515
(813) 441-3490

Attorneys, accountants, consultants. The services you perform and the advice you dispense are your firm's principle assets. How much of your time goes unaccounted for? How much time do you spend accounting for how much time you have spent?

Over the past eighteen months, Computer Consulting Services has developed, perfected and field-tested a powerful clerical and management tool called the C.C.S. Professional Timekeeping System. With an emphasis on quality and professionalism, the Timekeeping System offers the managers and decision makers of the firm a cost-effective tool which not only provides valuable management information, but also simplifies and optimizes the timekeeping and billing process, thereby increasing the productivity and profitability of the firm.

It's about time someone produced a quality timekeeping and billing system for the Apple. C.C.S. is now making this software generally available. Produced for professionals by professionals. See your local Apple dealer for details.

Hardware Requirements: APPLE II PLUS (48k), DOS 3.3
Dual disk drives, printer

Dealer Inquiries Invited

Master functions:

— Data entry
— Billing (pre-bill, regular bill, re-bill)
— Print data files
— Print activity analyses - entire firm
— Print activity analyses - single professional
— Edit data files
— Edit professional summary files
— Generate complete monthly client financial summaries
— Print complete monthly client financial summaries
— Print single client financial summaries

Utility functions:

— Edit parameter files
— Data file maintenance:
  Disk-to-disk transfer of client files
  Delete client files
  Re-start (compress) client files
— Print client data disk directories
— Initialize client data disks
— Backup client data disks

Capacities:

Number of Professionals	25
Number of clients (entire system)	UNLIMITED
Number of clients per disk	40
Number of data entries per disk	approx. 1600
Client file size	150 records

Select a modem for your computer ... easy as ABC.

CHOOSE A. The Smart One™, Model AM232, with its own built in microcontroller. Auto Dials, Auto Answers, Auto Modes (originate/answer), "Direct Connect" for *any* computer with an RS232 interface . . . **$299.00.**

CHOOSE B. The Acoustic Modem, Model AMII/A, for the Apple*. Comes complete with a powerful operating system. Supports 25 easy to use keyboard commands and operates interactively with your programs. No interface card required! . . . **$179.00.**

CHOOSE C. The "Direct Connect" Modem, Model AMII, for the Apple*. Comes complete with a powerful operating system. Supports Auto Dial, Auto Answer, 27 easy to use keyboard commands and operates interactively with your programs. No interface card required! . . . **$289.00.**

Be a part of the information revolution. It's fun, exciting, informative, educational. Call or write your order today. COD's accepted.

MICROMATE ELECTRONICS, INC.

2094 Front Street • East Meadow, New York 11554 • Telephone 516.794.1072

All "Direct Connect" modems are FCC certified. *Registered trademark of Apple Computer, Inc

BREAK THE TIME BARRIER!

Speed-up and expand Apple][functions with *TWO* all-new, low-cost utilities. . . *

1. UNIVERSAL BOOT INITIALIZER (UBI) **$49.95**

• UBI-initialized disks load RAM Card up to 9 times faster. • Applicable to Apple DOS 3.3 or 3.2.1 disks. • Obsoletes use of Apple BASICS disk and DOS 3.3 Master disk for fast single-disk direct booting. • The DOS 3.2.1 UBI-initialized disk boots on either 13 or 16 sector PROMS and loads RAM Card in 1.7 seconds (features not available from Apple). • Turnkey operation with boot in any Slot, Drive 1 and execution of BASIC, Binary or EXEC greeting program in Drive 1 or 2.

2. AMPER-SORT/MERGE (A-S/M) **$49.95**

• Can sort 1000 records in seconds. • General purpose Disk Sort/Merge Utility for Apple DOS 3.3 text files. • M/L Read, Sort and Merge for speed. • Sorts up to 5 input files into 1 output file. • Sorts random or sequential files alphanumerically (Ascending or Descending) up to 5 fields. • Merges 2 to 5 pre-sorted files into a single file. • Compatible with most data base programs that create standard DOS 3.3 text files (e.g., Personal Software's CCA DMS files).

SYSTEM REQUIREMENTS: 48K Apple][or][+, ROM or RAM Card, DOS 3.3 (or DOS 3.2.1 for UBI) and one or more disk drives.

COST: $49.95 (+$3.00 Postage and Handling) for each Utility Disk (with Training Disk and Manual).

FOR ONLY $15: You can get the Training Disk and Instruction Manual for either utility.

CREDIT CARDS: [VISA, MC or AMER EXP] Call **cybertronics international, inc.** Clearinghouse: (212) 532-3089.

MAIL ORDER: Send check (no Purchase Orders or C.O.D.'s) to S&H Software, Box 5, Manvel, ND 58256.

***LICENSED FROM APPLE:** *All Apple Computer Inc. programs — FPBASIC, INTBASIC, DOS 3.3 & DOS 3.2.1 — used in S & H Software's utility programs are licensed from Apple Computer Inc. by S & H Software. The unique UBI license to software vendors allows their use of these Apple Computer copyrighted programs to execute their programs on UBI-created disks, without further licensing from Apple Computer Inc. Software vendors' inquiries invited for both utilities.*

S&H Software

Box 5 Manvel ND 58256
(701) 696-2574

THE BUG

A sophisticated, yet easy to use diagnostic aid for getting "the bugs" out of your assembly language programs.

If you are a novice just getting started with assembly-language programming, you will find The BUG helpful in developing your understanding of how the Apple's 6502 internal processor operates. The many display options of The BUG will permit you to try out your assembly-language programs at the speed that is most comfortable for you. The BUG will also make it easy for you to see the effect of your program on the Apple as it executes.

If you are a professional programmer, you will also find that The BUG can improve your efficiency by reducing the time you spend identifying and solving complex, assembly-language programming errors. You will particularly appreciate the fact that The BUG offers the easiest to use and most extensive breakpointing capability of any "debugger" available for the Apple. Up to 13 different breakpoints can be specified to halt program execution when either: 1) a particular program location is reached, 2) one of the 6502 registers reaches a specified value, or 3) one of the bits in the 6502 status register reaches a specified value.

Another key feature of The BUG that serious programmers will appreciate is the ability to AUTOMATICALLY run lower-level subroutines at FULL SPEED. You no longer have to keep debugging the portions of your program that you already have working.

This is not the least expensive "debugger" program for the Apple, but we challenge you to find more capability for less money!

The BUG is supplied with a 40+ page user guide and is designed for use with DOS 3.3 on either the Apple II or Apple II Plus computer only **$50.00**

IMAGE PRINTER SERIES

Sensible Software is proud to introduce our new series of high resolution screen dumps. IMAGE PRINTERS provide a simple way to transfer high resolution graphic images onto paper. Each program in the series has unique features that give you full control of the printing. Some of the included options are:

- Full control over the area of the HIRES screen to be printed. You graphically pick the area for the utmost ease and accuracy.
- One-step printout of the picture with the ability to pause or abort the printing at any time.
- Menu-driven. All options are invoked with single keystrokes. IMAGE PRINTERS are extremely easy to use.
- Multiple image sizes, 6 different sizes for letter quality printers, 4 sizes for other printers.
- Creation of an inverse (negative) image for reverse printing.
- The ability to save the compressed and inverse images to disk.
- One time configuring for your printer and interface card. Why answer all those questions about your printer each time you want to print a picture?
- The images may be printed anywhere on the page.
- IMAGE PRINTERS support most popular interface cards, such as cards from Apple, California Computer Systems, Epson, and Mountain Computer. (The SSM AIO Serial Card and user-written 'driver' routines may be used with the letter quality printers.)

There are three separate versions of IMAGE PRINTERS, each one tailored to take full advantage of a different printer.

IMAGE PRINTER—LETTER QUALITY. For all popular letter quality printers (Diablo, NEC, Qume, etc.)

IMAGE PRINTER—EPSON. For the popular Epson MX-70, MX-80 and MX-100.

IMAGE PRINTER—NEC PC-8023A. For the NEC dot-matrix printer.

All versions are available for $40.00 ea.
Please specify version desired.

BUILD USING

. . . . provides an easy to use print-using routine plus similar functions for strings. Creating charts, reports and general screen formatting becomes a simple task. BUILD USING is written entirely in machine language and provides a simple means of avoiding garbage collection (those unnecessary delays that slow down your programs). With BUILD USING, you can choose how many digits should be displayed to right and left of the decimal point, and you can even fill the leading positions with the character of your choice. For example, you can print the number '157.23' as '157.2', or '0000157.230', or '*****$157. AND 23/100 DOLLARS', or hundreds of other ways (including exponential formats). Working with strings is just as easy; it's a snap to convert names from 'John' and 'Doe' to 'Doe, J.'. Also included are three levels of error trapping, so you can trap and correct numbers or strings that cannot fit in your specified format.

Utilities like BUILD USING are usually difficult to use because they must be located in one memory location (usually between DOS and the DOS file buffers); they cannot be used with your favorite editor or other special routines. BUILD USING does not have this limitation, as it can be easily located in many different memory locations: 1) the "normal" between DOS and DOS file buffers, 2) at HIMEM, 3) APPENDED to your Applesoft program, or 4) anywhere else in memory. Appending BUILD USING to your program is as simple as EXECing a TEXT file. BUILD USING uses the "CALL" command thereby leaving the ampersand vector free for your own use.

BUILD USING requires Applesoft in ROM (Language cards are find), DOS 3.3 and a minimum of 32K only **$30.00**

Sensible Software
6619 Perham Drive Dept. C
West Bloomfield, Michigan
48033 • (313) 399-8877
Visa and Mastercard Welcome
Please add $1.25 postage and handling per diskette

```
:ASM
                    ********************************
                    *                              *
                    * MX100 INITIALIZE TO 80 COLS. *
                    *      by  B. C. Detterich     *
                    *                              *
                    * Call -A.P.P.L.E. * Mar. 1982 *
                    *                              *
                    ********************************
                    *                              *
                    * This routine initializes to  *
                    * 80 cols without initializing *
                    * card, and restores normal    *
                    * character mode first, so it  *
                    * can be called anytime, and   *
                    * can be BLOADed anywhere. It  *
                    * does NOT turn off the screen;*
                    * use the usual Ctrl I 80N.    *
                    *                              *
                    ********************************
                    *
                    *
                    ESCAPE   EQU  155          ;Esc with hi bit set
                    CSWL     EQU  $36
                    PSLOTAD  EQU  $C102        ;Printer slot 1 address+2
                    COUT     EQU  $FDED
                    COUT1    EQU  $FDF0
                    RESTORE  EQU  $FF3F
                    SAVE     EQU  $FF4A
                    *
                             ORG  $300
                    *
                    *
0300: 20 4A FF               JSR  SAVE         ;Call register save
0303: A9 02                  LDA  #<PSLOTAD    ;Set output to printer
0305: 85 36                  STA  CSWL
0307: A9 C1                  LDA  #>PSLOTAD
0309: 85 37                  STA  CSWL+1
030B: A9 9B                  LDA  #ESCAPE
030D: 20 ED FD               JSR  COUT
0310: A9 46                  LDA  #'F'
0312: 20 ED FD               JSR  COUT
0315: A9 12                  LDA  #18          ;Ctrl R
0317: 20 ED FD               JSR  COUT
031A: A9 14                  LDA  #20          ;Ctrl T
031C: 20 ED FD               JSR  COUT
031F: A9 9B                  LDA  #ESCAPE
0321: 20 ED FD               JSR  COUT
0324: A9 51                  LDA  #'Q'
0326: 20 ED FD               JSR  COUT
0329: A9 50                  LDA  #80          ;80 Columns
032B: 20 ED FD               JSR  COUT
                    *
032E: A9 F0                  LDA  #<COUT1      ;Restore output to screen
0330: 85 36                  STA  CSWL
0332: A9 FD                  LDA  #>COUT1
0334: 85 37                  STA  CSWL+1
0336: 20 3F FF               JSR  RESTORE      ;Restore registers
0339: 60                     RTS

--END ASSEMBLY--

ERRORS: 0

58 BYTES
```

MX 100 HI-RES DUMP

```
:ASM

                         ********************************
                         *                              *
                         * MX100 INITIALIZE TO 80 COLS. *
                         *      by  B. C. Detterich     *
                         *                              *
                         * Call -A.P.P.L.E. * Mar. 1982 *
                         *                              *
                         ********************************
                         *                              *
                         * Initializes card to 80 cols. *
                         * Use for printer start-up;    *
                         * assumes normal chr. size mode*
                         *                              *
                         ********************************
                         *
                         *
0036:                    CSWL     EQU  $36
C100:                    PSLOTAD  EQU  $C100       ;Printer slot 1 address
FDED:                    COUT     EQU  $FDED
FDF0:                    COUT1    EQU  $FDF0
FF3F:                    RESTORE  EQU  $FF3F
FF4A:                    SAVE     EQU  $FF4A
                         *
                                  ORG  $300
                         *
                         *
0300: 20 4A FF           NXTCHR   JSR  SAVE        ;Call register save
0303: A9 00              NXTCHR   LDA  #<PSLOTAD   ;Set output to printer
0305: 85 36              NXTCHR   STA  CSWL
0307: A9 C1              NXTCHR   LDA  #>PSLOTAD
0309: 85 37              NXTCHR   STA  CSWL+1
030B: A2 00              NXTCHR   LDX  #$0
030D: BD 26 03           NXTCHR   LDA  CHAR,X      ;Get next character
0310: 20 ED FD           NXTCHR   JSR  COUT        ;Print it
0313: E8                 NXTCHR   INX
0314: EC 25 03           NXTCHR   CPX  NSTRING     ;Last character?
0317: D0 F4              NXTCHR   BNE  NXTCHR      ;No, loop back
                         *
0319: A9 F0              NXTCHR   LDA  #<COUT1     ;Restore output to screen
031B: 85 36              NXTCHR   STA  CSWL
031D: A9 FD              NXTCHR   LDA  #>COUT1
031F: 85 37              NXTCHR   STA  CSWL+1
0321: 20 3F FF           NXTCHR   JSR  RESTORE     ;Restore registers
0324: 60                 NXTCHR   RTS
                         *
0325: 07                 NSTRING  DFB  7           ;Nbr of chrs in data string
                         *
                         * Data string contains Ctrl I, 8, 0, N,
                         * Esc, Q, P; all with MSB off
0326: 09 38 30
0329: 4E 1B 51
032C: 50                 CHAR     DFB  9,56,48,78,27,81,80
                         *
                         * Change NSTRING and CHAR for new command line
--END ASSEMBLY--           ERRORS: 0        45 BYTES
```


For A.P.P.L.E. Members Only

The Power of VisiCalc®

by

Robert E. Williams and Bruce J. Taylor

THE POWER OF VISICALC is a book of exercises designed especially for users and potential users of the VisiCalc computer program. By performing simple step-by-step exercises, you will rapidly gain an ability to utilize the broad range of VisiCalc capabilities.

Better than an instruction book, THE POWER OF VISICALC demonstrates the use of VisiCalc features through specific application samples.

It will show you how to expand your use of VisiCalc, no matter what your application. Anyone with a problem that can be solved using a computer, will find THE POWER OF VISICALC an invaluable companion to their VisiCalc program.

No special training is needed to benefit from the exercises. All instructions are in plain English. The logic of each step is clearly spelled out, so you can later apply the information to your specific needs. THE POWER OF VISICALC will become your most valuable reference book as you expand your use of VisiCalc.

Member Price: $8.50

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055

VisiCalc is a registered trademark of Personal Software, Inc.

Overseas Shipping Wt. 10 oz.

Washington state residents add 6.6% sales tax.

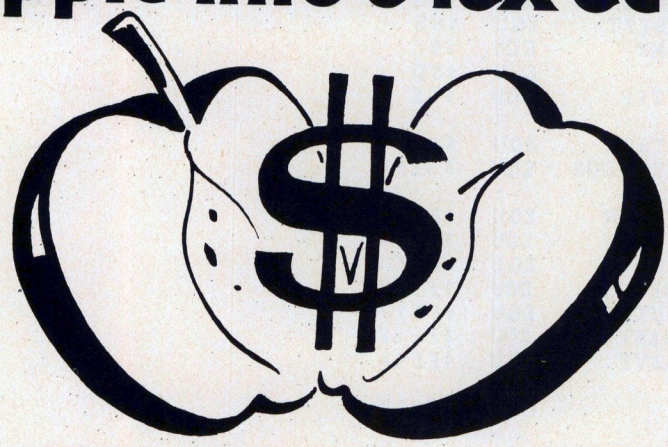
All foreign orders (incl. Canada) will require additional postage and shipping.

```
:ASM
                    ********************************
                    *                              *
                    * MX100  GRAPHICS DUMP ROUTINE *
                    *      by  B. C. Detterich     *
                    *                              *
                    * Call -A.P.P.L.E. * Mar. 1982 *
                    *                              *
                    ********************************
                    *
                    *
                    ESC       EQU  27
                    *
                    AKDLY     EQU  $A
                    USR       EQU  $B
                    SAVE26    EQU  $3C
                    SAVE27    EQU  $3D
                    SLOT      EQU  $3E
                    BYTE      EQU  $8A
                    SHIFT     EQU  $8B
                    INV       EQU  $8C
                    H         EQU  $8D
                    STEP      EQU  $8E
                    *
                    STACK     EQU  $100
                    TELLDOS   EQU  $3EA
                    *
                    HPOSN     EQU  $F411
                    INCRY     EQU  $F504
                    COUT      EQU  $FDED
                    SETVID    EQU  $FE93
                    OUTPRT    EQU  $FE97
                    RESTORE   EQU  $FF3F
                    SAVE      EQU  $FF4A
                    *
                              ORG  $9000
                    *
                    *
9000: 20 4A FF      START     JSR  SAVE
9003: 78                      SEI
9004: 20 58 FF                JSR  $FF58        ;Trick to push RTS on stack
9007: BA            REF       TSX
9008: BD 00 01                LDA  STACK,X
900B: A8                      TAY               ;Y=PCH
900C: CA                      DEX
900D: BD 00 01                LDA  STACK,X      ;A=PCL
9010: 18                      CLC
9011: 69 C6                   ADC  #PUNT-REF+1
9013: 85 0B                   STA  USR
9015: 98                      TYA
9016: 69 00                   ADC  #0
9018: 85 0C                   STA  USR+1
901A: A9 4C                   LDA  #$4C
901C: 85 0A                   STA  AKDLY
901E: 58                      CLI
901F: A5 3E                   LDA  SLOT
9021: 0A                      ASL
9022: 0A                      ASL
9023: 0A                      ASL
9024: 0A                      ASL
9025: 85 3F                   STA  SLOT+1
9027: A9 1B                   LDA  #ESC
9029: 20 0A 00                JSR  AKDLY
902C: A9 41                   LDA  #'A'
902E: 20 0A 00                JSR  AKDLY
9031: A9 08                   LDA  #8
9033: 20 0A 00                JSR  AKDLY
9036: A9 00                   LDA  #0
9038: 85 E2                   STA  $E2
903A: A9 28         NXTV      LDA  #40          ;Start new horiz line
903C: 85 8D                   STA  H
903E: A2 00                   LDX  #0
9040: A0 00                   LDY  #0
9042: A5 E2                   LDA  $E2
9044: 20 11 F4                JSR  HPOSN        ;Int Cur = Ext Cur
9047: A5 26                   LDA  $26          ;Save int cur on stack
```

micro lab presents a

TAXSAVER

Buy the TAX·MANAGER and turn your apple into a tax deduction.



Let the Tax Laws work for you - buy the TAX·MANAGER and deduct the Apple and the Program

Use The Tax-Manager to help you prepare your taxes. Then include the cost of this program and part of the cost of your computer as a deduction.

You don't have to be an accountant to understand it. Simply enter your tax information into this easy-to-use program. Then sit back and relax while it quickly computes the information and prints most of your federal income tax schedules. Get The Tax-Manager now and start the new year with your taxes under control.

When tax laws change, don't be concerned. The Tax-Manager will never be outdated. With Micro Lab's Extended Warranty Plan ($30 a year) you can update to the current version at no additional cost. You will not have to reinvest more money in a new program. The Tax-Manager, which is the first of the Micro Lab Tax Saver group, will keep you current.

THE TAX-MANAGER IS NOW AVAILABLE AT THE INTRODUCTORY PRICE OF $150.

© 1981, Micro Lab, Inc.

Apple is a trademark of Apple Computers, Inc.

micro lab

systems that work

2310 Skokie Valley Road
Highland Park, IL 60035 • 312-433-7550

Your personal income tax report, made easy...

micro lab TAX• MANAGER

FOR APPLE II 48K

- **Computes and prints out the 1040 form**
- **15 supporting forms and schedules**
- **Yearly updates of the program**
- **Automatic global recalculations of entries; recalculations linked from schedule to schedule, schedule to form.**

Don't get confused about which supporting forms and schedules you should file — use Tax-Manager.

Avoid paying penalties on income you didn't know you should have reported — use Tax-Manager.

Don't even think of the days necessary for compiling, calculating, checking and rechecking — use Tax-Manager.

Be sure your taxes are calculated with the right tax table and rate schedule — use Tax-Manager.

Don't get depressed about having to spend hours copying figures from work sheets to forms — use Tax-Manager.

Manage your personal tax situation to your best advantage — use Tax-Manager.

It is more than just a computer-aided, tax filing system and goes far beyond calculating taxes and preparing and printing the returns. It consists of a Tax Guide, Tax Checklist and Tax Preparation that, together, give you a complete tax management system.

This program is user-friendly and comes with a well-written manual so that even computer novices will find working with Tax-Manager easy. Our introductory price is $150 and it's deductible.

Apple is a trademark of Apple Computers, Inc.

micro lab

micro-lab for smart working software

2310 Skokie Valley Road • Highland Park, IL 60035 • 312-433-7550 **800/323-9083**

See your Micro lab dealer.

ALABAMA
Huntsville
Anderson Computers
Mobile
Computerland/Mobile

ARIZONA
Phoenix
Computerland/Phoenix
Scottsdale
Desert Computer

CALIFORNIA
Bakersfield
Burbank Computer World/Burbank
Cerritos
Byte Shop/Cerritos
Chatsworth
Opportunities, Inc.
El Toro
Wabash Apple Software
Fresno
National Computer Ctr, Inc.
Hayward
Computerland/Hayward
Huntington Beach
Sun Computers
Los Altos
Computerland/Los Altos
Malibu
Malibu Microcomputing
Mt. View
Microage Computer Store
Newport Beach
Computerland
Palmdale
Computers Made Easey
Sacramento
Capitol Computer Systems
San Jose
Computerland/San Jose
Small Business Assistance
Santa Clara
Affordable Computer Sys.
Santa Monica
Apple Cyber
Santa Rosa
Santa Rosa Computer
Stockton
Nanci's Software
Torrence
Net Profit Computers
Van Nuys
Computer World
Yucaipa
Yucaipa Computer System

COLORADO
Boulder
Computer Connection
Colorado Springs
Computer Mart
Computerland/Col Springs
Evergreen
Computer Video Concepts

CONNECTICUT
Hamden
Computerland/New Haven
New London
Computer Lab
Norwalk
Micro Computer Store
Southington
A M Computer Products
Westport
Computerworks
Windsor Locks
The Computer Store

WASHINGTON DC
Washington
The Program Store

FLORIDA
Boca Raton
Computerland/Boca Raton
Clearwater
Computer Work
Cocoa Beach
Creative Computer Cervice
Coral Gables
International Computer
Ft. Lauderdale
Franklin Ed. Serv.
Ft. Meyers
Computerland/Ft. Meyers
Ft. Walton Beach
Tampico Lab Inc.
Gulf Breeze
Computer Store/Gulf
Holly Hills
Mini Concepts
Naples
The Other Phone Store
Palm Beach
Computerland/Palm Beach
Palmetto
Natural Systems Computers
Plantation
Turn Key Mini Computer
Sarasota
Computerland/Sarasota
West Palm Beach
Computer Center/Palm Beach

GEORGIA
Symrna
Computerland/Atlanta

HAWAII
Kailua
Computer Mkt.

IOWA
Ames
Beacon Electronics
Bettendorf
Memory Bank Inc.
Des Moines
Computer Emporium, Inc
Sioux City
Pflanz. Electronics
Waterloo
Computer Center

IDAHO
Boise
Northwest Computer Center
Idaho Falls
R & L Data Systems
Yost Office

ILLINOIS
Chicago
Computer Room
Erickson Communications
Multi Media Education
Decatur
Main Street Computer Co.
Deerfield
Video Etc.
Edwardsville
Computer Corner
Elmhurst
Computer Junction
Evanston
Nabihs
Glen Ellyn
Computer Center
Morton Grove
Compushop
Naperville
Illini Microcomputers
Niles
Computerland/Niles
North Brook
Northbrook Computer
Oak Park
Bies. Systems Inc.
Oaklawn
Computerland/Oaklawn
Rockford
Alpine Computer Center
Rolling Meadows
Compushop/Rolling Meadows
Schaumburg
Data Domain/Schaumburg

INDIANA
Bloomington
Data Domain/Bloomington
Merrillville
Computerland/Merrillville
Munster
Meade's Video Comp. Center
South Bend
General Microcomputer

KANSAS
Overland Park
Computerland/Overland
Personal Computer Center
Reading
Midwest Research Con.
Topeka
Computerland/Topeka

MASSACHUSETTS
Burlington
Computer Store
Charleston
Computer City
East Longmeadow
The Small Computer Co.
Sudbury
The Computer Store/301
The Computer Store/Mclean
Watertown
Microcon Software Ctrs.

MARYLAND
Annapolis
Computers Etc.
Baltimore
Muse Computer Center
Columbia
Adca
Rockville
Computerland/Gaithersburg
Severna Park
The Computer Forum

MAINE
Portland
Harper Electronics

MICHIGAN
Kentwood
Computerland/Grand Rapids
Willman
Tri-State Digital Prod.

MINNESOTA
Marshall
Slosman & Associates
Minneapolis
Micronage Computer Store
Personal Business Systems
Team Central Inc
Team Central Inc.
St. Paul
Schaak Electronics
Imtech, Inc.

MISSOURI
Clayton
Forsyth Computers
Kansas City
Computer Work Shop
Springfield
Computer Mart
St. Charles
United Computer
St. Louis
Carleton Agency Co.
Futureworld Inc.

MISSISSIPPI
Jackson
Mississippi Micros, Inc.

NEBRASKA
Lincoln
Alcom
Gering
Pan Handle Tech
Kearney
Computer Hardware
Omaha
Abacus, Inc.
Computerland/Omaha

NORTH CAROLINA
Burlington
Nova Scientific Corp.
Charlotte
Computer Room Inc.
Standard Computer
Greensboro
Computer Shop

NORTH DAKOTA
Grand Fork
Team Electronics
Minot
Team Electronics

NEW HAMPSHIRE
Portsmouth
The Think Tank

NEW JERSEY
Cherry Hill
Computerland/Cherry Hill
Englewood
American Business Product
Englishtown
Computer Madness, Inc.
Marlton
Midas Data Systems
Morristown
Computerland/Morristown
North Plainfield
Boise Office Equip.
Paramus
Computer Universe
River Edge
Software City
Summit
Stonehenge Computer Shop

NEW MEXICO
Alamogordo
Personalized Comp. Serv.

NEVADA
Las Vegas
Century 23

NEW YORK
Albany
Computer Room
Buffalo
Comuterland/Buffalo
Carle Place
Computerland/Nassau
Garden City
The Computer Factory, Inc.
Kingston
Computer Shop
Melville
Future Vision

New Hyde Park
Berliner Computer Center
New Rochelle
Calculator & Computer Center
New York
Computer Era
Datel
Morris Decison Systems
Computer Factory
Pleasantville
Micro Center
Sunburst Communications
Rochester
The Computer Center
Syracuse
Omnifax Computer Stores
Wappinger Falls
Mr. Computer

OHIO
Cincinnati
21st Century Shop
Computerland/Cincinnati
Columbus
Microage

OKLAHOMA
Lawton
Computer Tech, Inc.
Tulsa
Computerland/Tulsa

OREGON
Creswell
Computertutor
Eugene
Custom Computers

PENNSYLVANIA
Allentown/Whitehall
Computerland
Dresher
Computerland/Dresher
Erie
Erie Computer
Mechanicsburg
Computerland/Harrisburg
Monroeville
Computer Workshoppe
Paoli
Computerland/Paoli
Philadelphia
Caldwell Computer Corp.
Pittsburgh
The Computer House
Whitehall-Allentown
Computerland/Lehigh

RHODE ISLAND
Providence
The Computer Store

SOUTH CAROLINA
Rockhill
Gismo Computer

TENNESSEE
Memphis
Computer Lab
Opus II

TEXAS
Abilene
Computer Shop/Abilene
Arlington
Computer Port
Arlington
Computer Port
Austin
The Computer center
Computer 'N Things
Beaumont
Computer Concepts
College Station
Young Electronics
Dallas
Computeria
Houston
Computer Center
Computer City/Houston
Computercraft/Houston
Marshall
Marshall Business Corp.
Odessa
Computer Patch
Richardson
Compushop
Micro Age Computer
San Antonio
Computer Solution
R. L. Coles Electric
The Computer Shop
Tyler
Computerland/Tyler
Waco
Computer Solution Inc.
Wichita Falls
The Computer Store

UTAH
Ogden
Computer Specialists
Salt Lake City
Computer Solutions

VIRGINIA
Springfield
Computers, Etc.

WASHINGTON
Bellingham
Western Micro Computer C.
Federal Way
Software Serv.
Port Angeles
Angeles Computer, Inc.
Seattle
Empire Electronics Inc.
Spokane
Personal Comp. Inc.
Yakima
Cliff Miller's Inc.

WISCONSIN
Appleton
Computer World/Appleton
Janesville
Team Electronics
Lacrosse
Midwest Computer
Madison
Computerland/Madison
Laboratory Consulting, Inc.
Milwaukee
Comptuerland/Milwaukee
Computers Plus
North Shore Computers Inc.
Team Electronics/Milwaukee

WYOMING
Jackson
Foxfire Computers

© 1981, Micro Lab, Inc.

MX 100 HI-RES DUMP

```
9049: 85 3C      76          STA  SAVE26
904B: A5 27      77          LDA  $27
904D: 85 3D      78          STA  SAVE27
904F: A9 1B      79          LDA  #ESC
9051: 20 0A 00   80          JSR  AKDLY
9054: A9 4B      81          LDA  #'K'
9056: 20 0A 00   82          JSR  AKDLY
9059: A9 18      83          LDA  #24
905B: 20 0A 00   84          JSR  AKDLY
905E: A9 01      85          LDA  #1          ;256
9060: 20 0A 00   86          JSR  AKDLY
9063: A9 00      87  NXTH    LDA  #0          ;Next hor screen byte
9065: 85 8B      88          STA  SHIFT
9067: A9 08      89  NXTBIT  LDA  #8          ;Next bit in screen byte
9069: 85 8E      90          STA  STEP
906B: A4 E5      91  NXTSTEP LDY  $E5         ;Next vert dot
906D: B1 26      92          LDA  ($26),Y
906F: A4 8B      93          LDY  SHIFT
9071: 6A         94  SHLOOP  ROR
9072: 88         95          DEY
9073: 10 FC      96          BPL  SHLOOP
9075: 26 8A      97          ROL  BYTE
9077: 20 04 F5   98          JSR  INCRY       ;INCRY Int Cur
907A: C6 8E      99          DEC  STEP
907C: D0 ED      100         BNE  NXTSTEP
907E: A5 8A      101         LDA  BYTE
9080: 45 8C      102         EOR  INV         ;Inv print mask
9082: 20 0A 00   103         JSR  AKDLY
9085: A5 3D      104         LDA  SAVE27      ;Restore int vert cur
9087: 85 27      105         STA  $27
9089: A5 3C      106         LDA  SAVE26
908B: 85 26      107         STA  $26
908D: E6 8B      108         INC  SHIFT
908F: A9 07      109         LDA  #7
9091: C5 8B      110         CMP  SHIFT
9093: D0 D2      111         BNE  NXTBIT
9095: E6 E5      112         INC  $E5
9097: C6 8D      113         DEC  H
9099: D0 C8      114         BNE  NXTH        ;Not done w hor line
909B: A9 80      115         LDA  #128        ;Done, so flush buffer
909D: 85 8D      116         STA  H
909F: A9 08      117 FLOOP   LDA  #8          ;Ctrl H
90A1: 20 0A 00   118         JSR  AKDLY
90A4: C6 8D      119         DEC  H
90A6: D0 F7      120         BNE  FLOOP
90A8: A9 0A      121         LDA  #10         ;Ctrl J (line feed)
90AA: 20 0A 00   122         JSR  AKDLY
90AD: A9 E2      123         LDA  #$E2
90AF: 18         124         CLC
90B0: 69 08      125         ADC  #8
90B2: 85 E2      126         STA  $E2
90B4: C9 C0      127         CMP  #"@"
90B6: D0 82      128         BNE  NXTV        ;More picture, next hor line
90B8: A9 1B      129         LDA  #ESC
90BA: 20 0A 00   130         JSR  AKDLY
90BD: A9 32      131         LDA  #'2'        ;For 6 LPI
90BF: 20 0A 00   132         JSR  AKDLY
90C2: 20 93 FE   133         JSR  SETVID      ;Back to screen
90C5: 20 EA 03   134         JSR  TELLDOS
90C8: 20 3F FF   135         JSR  RESTORE
90CB: 60         136         RTS
                 137 *
90CC: 85 8A      138 PUNT    STA  BYTE
90CE: 20 97 FE   139         JSR  OUTPRT
90D1: A9 89      140         LDA  #$89
90D3: 20 ED FD   141         JSR  COUT
90D6: A9 CE      142         LDA  #"N"
90D8: 20 ED FD   143         JSR  COUT
90DB: A4 3F      144         LDY  SLOT+1
90DD: A5 8A      145         LDA  BYTE
90DF: 99 80 C0   146         STA  $C080,Y     ;Direct to printer
90E2: 60         147         RTS

--END ASSEMBLY--      ERRORS: 0          227 BYTES
```

```
]LIST
10  REM

MX100 Command Demo  B C Detterich

  Call -A.P.P.L.E. * Mar. 1982 *

20  REM

Program demo of machine language
initialization methods.

Sequence:
  1. Power up printer
  2. Run this program.

30  REM

This program will:
 Initialize to 80 columns with
  MX100 example
 Print a line > 80 columns to
  show auto return was generated
 Issue PR#0 which normally
  re-enables screen echo
40  REM
POKE $C102 to CSW to bypass card
  initialization
 Print 80 columns again to show
  there is no echo
 Deselct printer with PR#0
 Series of POKEs without card
  enabled to set 30 columns, set
  compressed characters
50  REM
 POKE CSW with $C102 and print
  > 30 columns (note: still no
  screen echo and printer took
  commands without being enabled)

 Issue PR#0 to deselect
 BLOAD and CALL MX100-2, not at
  $300 (768) !!
60  REM
 Issue PR#1 and print >80 columns

 Normal characters, 80 columns
and echo are back.
```

```
*
]LIST

10  REM

Using  the MX100 Graphics Dump
from Applesoft  : B C Detterich

 Call -A.P.P.L.E. * Feb. 1982

100  HIMEM: 36000
110  PRINT  CHR$ (4)"BLOAD MX100
     GRAPHICS DUMP"
120  PR# 1: CALL 1002: REM Assume
      printer slot 1
130  POKE 140,0: REM  NORMAL; use
      255 for INVERSE
140  POKE 230,32: REM Hi-Res page
      1
150  CALL 36864: REM $9000
160  PR# 0: CALL 1002
```

```
500  PRINT  CHR$ (4)"BLOAD MX100-
     1"
510  CALL 768
520  PR# 1: CALL 1002: PRINT  CHR$
     (9)"136N"
530  GOSUB 1000: REM

600  PR# 0: CALL 1002
610  POKE 54,2: POKE 55,193
620  GOSUB 1000: REM

700  PR# 0: CALL 1002:BYPASS = 49
     296:ESC = 27:Q = 81:CO = 15
710  POKE BY,ESC: FOR I = 1 TO 10
     0: NEXT
720  POKE BY,Q: FOR I = 1 TO 100:
      NEXT
730  POKE BY,30: FOR I = 1 TO 100
     : NEXT
740  POKE BY,13: FOR I = 1 TO 100
     : NEXT
750  POKE BY,CO
760  POKE 54,2: POKE 55,193
770  GOSUB 1000: REM

800  PR# 0: CALL 1002
810  PRINT  CHR$ (4)"BLOAD MX100-
     2,A12000"
820  CALL 12000
830  PR# 1: CALL 1002
840  GOSUB 1000
850  PR# 0: CALL 1002
900  END : REM

1000  FOR J = 1 TO 10: FOR I = 0 TO
      9: PRINT I;: NEXT I,J: PRINT
      : RETURN
```

```
*9000.90E2

9000- 20 4A FF 78 20 58 FF BA
9008- BD 00 01 A8 CA BD 00 01
9010- 18 69 C6 85 0B 98 69 00
9018- 85 0C A9 4C 85 0A 58 A5
9020- 3E 0A 0A 0A 0A 85 3F A9
9028- 1B 20 0A 00 A9 41 20 0A
9030- 00 A9 08 20 0A 00 A9 00
9038- 85 E2 A9 28 85 8D A2 00
9040- A0 00 A5 E2 20 11 F4 A5
9048- 26 85 3C A5 27 85 3D A9
9050- 1B 20 0A 00 A9 4B 20 0A
9058- 00 A9 18 20 0A 00 A9 01
9060- 20 0A 00 A9 00 85 8B A9
9068- 08 85 8E A4 E5 B1 26 A4
9070- 8B 6A 88 10 FC 26 8A 20
9078- 04 F5 C6 8E D0 ED A5 8A
9080- 45 8C 20 0A 00 A5 3D 85
9088- 27 A5 3C 85 26 E6 8B A9
9090- 07 C5 8B D0 D2 E6 E5 C6
9098- 8D D0 C8 A9 80 85 8D A9
90A0- 08 20 0A 00 C6 8D D0 F7
90A8- A9 0A 20 0A 00 A9 E2 18
90B0- 69 08 85 E2 C9 C0 D0 82
90B8- A9 1B 20 0A 00 A9 32 20
90C0- 0A 00 20 93 FE 20 EA 03
90C8- 20 3F FF 60 85 8A 20 97
90D0- FE A9 89 20 ED FD A9 CE
90D8- 20 ED FD A4 3F A5 8A 99
90E0- 80 C0 60
```

WHAT'S EATING YOUR APPLE?

Now with these new features:

Linked Tests
6502 CPU Test
Printed Results
Disk Speed Test
Boots From Any Slot

FIND OUT WITH APPLE-CILLIN II™

It's a fact of computer life. Software and hardware interact. If there's a problem eating your Apple, you can spend hours trying to find out whether software or hardware is the culprit. Unless you have Apple-Cillin II.

Apple-Cillin II is a comprehensive diagnostic system developed by XPS, Inc. to check the performance of your Apple II or Apple II+ and its peripherals. Apple-Cillin II will verify over two dozen hardware operations, and either identify a specific problem or give your hardware a clean bill of health. Quickly and easily.

Apple-Cillin II works with 13 and 16 Sector drives. It is written entirely in assembly language and is compatible with any Apple II or Apple II+ with 48k of RAM and one or more disk drives.

To order Apple-Cillin II — and to receive a free catalog of our other innovative software — call us toll free at 800-233-7512. In Pennsylvania call 717-243-5373. Or send $49.95 (Pa. residents add 6% sales tax) to XPS, Inc., 323 York Road, Carlisle, PA 17013.

Apple II and Apple II+ are trademarks of Apple Computer, Inc.

XPS, Inc.

Quality Microcomputer Software
323 York Road, Carlisle, PA 17013
(800) 233-7512 / (717) 243-5373

CALL —A.P.P.L.E. CONTEST WINNNERS:

OUT OF SORTS

SUCH a response to our contest! We're flabbergasted! And such excellent entries, all contributing to the judges' dilemma. In the end we wound up by awarding multiple second prizes.

Scott Simpson of Fort Wayne, IN is our first prize winner, and a $200 cash prize will soon be on its way to him. In addition, he will receive our standard *Call* -A.P.P.L.E. page rate of $50/page for his explanation, which follows in its entirety.

Contest responses were international in scope, witness an excerpt from second prize winner Luis Arana of Buenos Aires, Argentina. Additional second prizes, consisting of $50 worth of club software, were awarded to: Scott Lee, David Walker, Peter Lincoln and Brian Matthews.

Both Lee and Arana proffered Assembly Language versions of the sort routine. We hope our readers will further explore these, and take particular note of the tremendous speed advantage that can be demonstrated by replacing a BASIC routine with an Assembly Language routine.

Author Tom Crawford also provides a few answers of sorts *in the following comments:*

1. The variable W stands for "Wun," and I used it in lines 310 and 410 strictly to make the program go faster. Why didn't I use it everywhere? I should have and just didn't. So what did you *expect?*
2. There once was a subroutine at line 1000 which printed the current contents of the screen as a 40 x 40 array of color values. Ed Jackson took advantage of that "hook" and installed a subroutine which counted each color after each sort was done, and sure enough, after each pass, the actual quantity of each color was the same. So, what *did* you expect?
3. What would I name the program? Seven of the 22 entries named it "Double Bubble" or something very close. A plurality of nearly one-third is certainly a winner. My favorites are "Gravity's Rainbow," submitted by Don O'Shea, and "Decreasing Entropy," suggested by J. G. LoBue.
4. This was a fun contest and I was astonished by the response. Who will write the next one that I might submit an entry?

DIETICIAN

A GREAT NEW APPLICATION FOR THE MICROCOMPUTER

- Make up a diet menu with specified content of calories, carbohydrates, proteins, fats, cholesterol and sodium.
- Program comes complete with the composition of 700 foods.
- Allows you to look at the composition of each food instantaneously before deciding to include it in your diet. our diet
- Make as many menus as you like, each one with your own choice of foods.
- Make a different menu for each day of the week or the month. Takes only minutes to make up a menu.
- Store the menus on the diskette or print them out.
- Analyze popular advertised diets
- Originally written for use in a physician's office. Now made available to the public.
- Requires Apple II plus or Apple II with Applesoft in ROM and one disk drive with DOS 3.3. Printer desirable.
- Comes complete with documentation.

Calories = 646 Carbohydrate = 85 Cholesterol = 250 Protein = 32 Sodium = 533 Fats = 23

	(GMS)		Cal	Carb	Prot	Fats	Chol	Sod
Juice-Orange	248	1 Cup	112	26	2	1	0	2
Egg-Boiled	57	1 Item	82	1	7	6	250	61
Toast-White	40	2 Slice	124	24	4	2	0	234
Coffee	240	1 Cup	4	1	0	0	0	2
Sugar-Granltd-Teaspoon	4	1 Teaspoon	15	4	0	0	0	0
Creamer-Powder	5	1 Teaspoon	10	1	0	1	0	0
Recipe *Bread-White Enr.	40	2 Slice	124	24	4	2	0	234

Please RUSH me _____ copies of DIETICIAN 1 for APPLE II at $59.95 each.

PAYMENT ENCLOSED $ . . . . . (VISA / MASTERCHARGE EXP. DATE _____

Add $2.00 for Postage and Handling ACCT. NO. _____

(Texas residents add 6% sales tax)

Name _____

Address _____

City _____ State _____ Zip _____

DIETWARE
P.O. Box 503
Spring, Texas 77373

APPLE II is the trademark of the APPLE COMPUTER Inc.

FOR YOUR APPLE II

Industry standard products at super saver discount prices

PARALLEL PRINTERS

C-ITOH 8510 PRINTER Virtually identical to the NEC 8023. Same electro-mechanical internals. The only difference is the name and the price. Specifications: • 100 CPS dot matrix printer • 80 column print—136 characters per line • Tractor/friction feed • 7 different print fonts included • 2K printer buffer • Proportional spacing • Bit image graphics and graphic symbols.

NEC & C-ITOH List: $800

Special Price with any other purchase **$595**

EPSON 100 PRINTER **$749**

Z-80 CARDS FOR YOUR APPLE

MICROSOFT SOFTCARD. With CP/M® & MBASIC.

(List: $399) **$289**

NEW! ADVANCED LOGIC SYSTEM Z-CARD. With C/PM®. Has everything the Softcard has except MBASIC. Works with Microsoft's disks too.

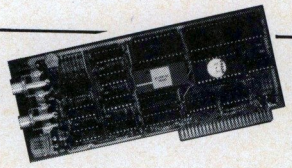
(List: $269) **$209**

NEW! U-Z-80 PROCESSOR BOARD (from Europe) Software compatible with Softcard and ALS Software ..... (List: $199) **$149**

With CP/M®, ... add: (List: $75) **$49**

NEW! ALS SYNERGIZER. CP/M™ operating package with an 80 column video board, CP/M® interface, and 16K memory expansion for Apple II. Permits use of the full range of CP/M® software on Apple II *and* Apple III.

(List: $749) **$595**



80 COLUMN VIDEO DISPLAYS FOR APPLE II

SMARTERM.
(Not to be confused with SUPRTERM)

Software switching from 80 to 40 and 40 to 80 characters. 9 new characters not found on the Apple keyboard. Fully compatible with CP/M® and Apple PASCAL®. With lowest power consumption of only 2.5 watts.

(List: $360) **$279**

SMARTERM EXPANDED CHARACTER SET. 7"x 11" matrix with true decenders. Add to above: **$40**

VIDEX-80 ..... **$279**

WIZARD **$269**

MONITORS FOR YOUR APPLE

AMDEX 300G **$229**

SUPER SPECIALS!

12" GREEN MONITOR **$139**

12" BLACK & WHITE MONITOR **$125**

All equipment shipped factory fresh. Manufacturers' warranties included. Please add $3.00 per product for shipping and handling. California: add 6% tax; BART Counties: 6½%.

EXPAND YOUR APPLE II MEMORY

16K MEMORY EXPANSION MODULE. The preferred 16K RAM Expansion Module from PROMETHEUS. Fully compatible with CP/M® and Apple Pascal®. With full 1-year parts and labor warranty.

(List: $169) **$99**

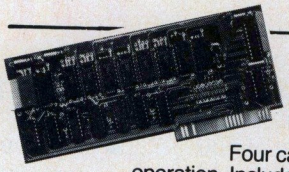
MEMORY MANAGEMENT SYSTEM. Provides more free memory in BASIC by moving DOS into 16K card.

(List: $49) **$35**

16K RAM ADD-ON KITS. Eight 16K full specification RAMS.

Reduced to **$19**

SATURN SYSTEMS 32K RAM CARD **$199**



VERSAcard from PROMETHEUS

Four cards on one! With true simultaneous operation. Includes: (1) Serial Input/Output Interface, (2) Parallel Output Interface, (3) Precision Clock/Calendar, and (4) BSR Control. All on one card. Fully compatible with CP/M® and Apple Pascal®.

(List: $249) **$195**

JOYSTICK

Takes the place of two Apple Paddle Controllers. From BMP Enterprises. Heavy duty industrial construction and cable. Non-self centering. With polarity switches for consistent motion control.

(List: $59) **$39**

MODEMS FOR YOUR APPLE II

HAYS Smartmodem **$229**

NOVATION Apple Cat **$349**

CENTRONICS COMPATIBLE PARALLEL INTERFACE

NEW! Look out Grappler! From PROMETHEUS. For use with Epson, NEC, C-ITOH, and other printers. Fully compatible with CP/M® and Apple Pascal®.

Only **$69**

With Screen Dump Graphics for EPSON printer, add ... **$25**

Phone for Quick Shipment!

(415) 490-3420

SGC

"The mail order specialists"

342 Quartz Circle, Livermore, CA 94550

HEXADECACOLOR DOUBLE-BUBBLE DEMO SORT

By Scott Simpson/1st Place Contest Winner

A Rainbow Bounce
Thru the Ups and Downs
(And the Backs and Forths)
Or a Two-Dimensional Bubble Sort

What's This About?

TOM Crawford posed the readership a crafty little challenge in the Sept., '81 issue ("So What Did You Expect Dept?" page 77).

First he presents us with a program listing (reproduced at the end of this article), and then invites us to key it in, run it, and tell him the following:

A.) What is the program doing?
B.) What do the changing numbers at the bottom of the screen mean?
C.) What will the finished Low-Res pattern look like?
D.) Will the program always work, or not?
E.) What would you call something that does what this thing does?

If you haven't seen this little program yet, JuMP to the end of this article, key it in and try it out. (Be aware that it runs for about a half hour, but it's intriguing to watch, and if you just let your mind "follow the bouncing ball," it's a soothing way to spend thirty minutes.)

After you've seen the program running, come back here and read my answers to Tom's five-question quiz!

What's It Doing?

It is using a display of Low-Res graphic blocks to visually demonstrate, through color and motion, the step-by-step workings of a two-dimensional Bubble Sort sorting routine. (And doing it very well, I might add! It is one of the clearest Bubble Sort demonstrations I've ever seen.)

The Bubble Sort is a method of sorting a list or array of data into ascending or descending order. Strong points: easy to program and easy to understand. Weak point: very, VERY slow. (Note that it takes around thirty minutes to sort 1600 items in this program!)

For those of you who haven't run into the Bubble Sort yet, the following line-by-line "step through" of Tom's program will illustrate the technique very nicely...

```
80  W = 1
```

This line sets the variable "W" equal to "1." This variable will be used later as a constant increment amount to "compare the next block." (If we're sitting on the block at screen position I,J, the next block to the right would be I+W,J ; the next block below I,J would be I,J+W.)

Using a variable as an increment (J+W) is faster than using a numerical constant (J+1), so this is a good programming technique for speed. (Why is it faster? Check page 120 in your Applesoft manual for a complete explanation of this and other program speed-up techniques.)

```
89  HOME
90  GR
```

A couple of "housekeeping" lines; the first clears the screen, and the second sets the Mixed-Text-and-Graphics mode.

```
100 FOR I = 0 TO 39
110 FOR J = 0 TO 39
120 C = INT ( RND (1) * 16)
130 COLOR = C
140 PLOT I,J
150 NEXT J
160 NEXT I
```

These seven lines form a loop that fills the 40 × 40 graphics screen with 1600 random colored blocks. Line 120 generates a random number (C) between 0 and 15; Line 130 uses that number to select one of the sixteen Low-Res graphic colors (Black = 0, Magenta = 1, ... Aqua = 14, White = 15); Line 140 plots the block at I,J in that randomly selected color. (Note that this builds the grid column-by-column across the screen. If you want to build it row-by-row down the screen, you need only change Line 140 to read: **PLOT J,I.)**

The next line is pure red herring...

```
170 GOSUB 1000
```

What's on the other end? Nothing! A multi-line remark, which Applesoft ignores, and a RETURN, which it doesn't. But, like the loyal old dog that it is, Applesoft will trot dutifully off to line 1000, sniff around for something to do, stumble over the RETURN at line 1001, and come trotting back to the main scent. No harm done, and after all, if you read the remark at Line 1000 ... Well, what *did* you expect? (There's an identical ruse at Line 490. Don't expect any more there.)

```
200 F = 0
210 V = 38
220 H = 39
```

Line 200 is the top of the sort routine main loop. This line sets variable "F" equal to zero each time we go through the main loop.

Lines 210 and 220 also set variables that will be used later as loop counters.

```
300 FOR I = 0 TO H
305 FOR J = 0 TO V
310 IF SCRN (I,J) =    SCRN
    (I,J + W) THEN 370
325 G = SCRN (I,J + 1)
330 COLOR = SCRN (I,J)
335 F = F + 1
340 PLOT I,J + 1
350 COLOR = G
360 PLOT I,J
370 NEXT J
380 NEXT I

400 FOR I = 0 TO H
405 FOR J = 0 TO V
410 IF SCRN (J,I) =    SCRN
    (J + W,I) THEN 470
425 G = SCRN (J + 1,I)
430 COLOR = SCRN (J,I)
440 PLOT J + 1,I
450 COLOR = G
460 PLOT J,I
470 NEXT J
480 NEXT I
```

HEXADECACOLOR DOUBLE-BUBBLE

There it is! The sort routine, itself. Two of them, in fact, back-to-back. The first sort loop runs from Line 300 to Line 380. This loop moves across the grid from left to right, scanning each column from top to bottom as it goes. Line 310 says, "If the color value of the block I'm standing on now is equal to or less than the color value of the next block below me, simply step onto that next block. (**THEN**[GOTO]**370**)

"However, if the color value of the block I'm standing on now is *greater* than the color value of the next block below me, store the value of that next block as "G" (Line 325). Change the plotting color to that of the block I'm standing on now (Line 330), and plot the box below me in that color (Line 340). Change the plotting color again, this time to the former color of the block below me, which we saved as "G" (Line 360), and THEN step onto the block below (Line 370)."

BYTES & PIECES

WOW! CHECK THESE PRICES

Apple Software • All Programs On Disk

WARP DESTROYER	29.95	*now*	23.96
DARK FOREST	29.95	*now*	23.96
RED ALERT	29.95	*now*	23.96
CYBORG	32.95	*now*	26.36
THRESHOLD	39.95	*now*	31.96
CROSSFIRE	29.95	*now*	23.96
HADRON	34.95	*now*	27.96
WIZARDRY	49.95	*now*	39.96
SPACE QUARKS	29.95	*now*	23.96
BUG ATTACK	29.95	*now*	23.96
TRACK ATTACK	29.95	*now*	23.96
TASC COMPILER	175.00	*now*	139.96
DAVID'S MIDNIGHT MAGIC	34.95	*now*	27.96
[*SPECIALS*]			
TIME ZONE	99.95	*now*	74.76
BEER RUN	34.95	*now*	26.76
ARCADE MACHINE	44.95	*now*	33.16
HOME ACCOUNTANT	74.95	*now*	57.96
HI RES ADVENTURE #4	34.95	*now*	26.76

C.O.D. • Money Orders • Certified Checks
For Personal Check Allow Two Weeks
N.Y.S. Res. Add 7.25% Sales Tax • All Orders Under $100 Add $2.00 Postage and Handling
• SEND FOR FREE PRICE LIST #650 •

BYTES & PIECES **(516) 751-2535**

Box 525 Dept. DB • E. Setauket, N.Y. 11733

This process compares each pair of blocks, and exchanges the colors of the blocks, if necessary, to insure that the blocks with the larger color values are "carried along" toward the bottom of the columns.

Repeat the same process for every block in that column, and then start over again at the top of the next column (Line 380).

The effects of this first sort loop are obvious when you watch the program in action. The smaller color values "bubble up" to the top of the columns, and the larger color values "sink" to the bottom of the columns. It is this "bubbling" action that gives the Bubble Sort its name.

The first sort routine (Lines 300-380) is followed by a second sort (Lines 400-480), which is essentially the same as the first, except that it turns things 90 degrees. Instead of moving across the screen, scanning the columns from top to bottom as it goes, the second sort moves down the screen, scanning the rows from left to right as it goes. Also, rather than "bubble" the smaller color values to the top and the larger ones to the bottom, the second sort "bubbles" the larger color values to the LEFT, and the smaller ones to the RIGHT.

Hint: compare both sorts line-per-line; it's the subtle changes in Line Pairs 310/410, 325/425, 330/430, 340/440, 350/450, and 360/460 that make the difference.)

Taken together, the two sort routines make up a complete two-dimensional sort cycle. While it's difficult to visualize the net effect of the whole cycle just by looking at the lines of code, it becomes clear when you watch the program in action.

Since the first sort tends to move the large color values down in the grid, while moving the small color values up ... and the second sort tends to move the large color values to the left as it moves the small color values to the right ... you can "add the two sets of effects together" and see that the net effects of one two-sort cycle are to move the large color values toward the lower-left, and the small color values to the upper-right.

```
500 IF F = 0 THEN STOP
```

Since Line 200 sets F = 0 each time before the sort routines begin and Line 500 stops the program if F still equals 0 after the sort routines have been run, the variable "F" must somehow tell us (and the computer) if the array of blocks has been completely sorted. It does, and here's how:

The Bubble Sort does its sorting by exchanging the colors of any two neighboring blocks that are "out of order." If no exchanges are made, then all the blocks must be in order. The variable "F" simply counts the number of exchanges that are made each time the complete two-cycle sort routine goes through the array. (Lines 335 and 435 increase F by one each time an exchange is made.)

So, if F = 0 at Line 500, no exchanges were made during the most recent pass through the sort routines, the array is therefore completely sorted, and the program stops. If F does NOT equal zero, this tells the computer that some exchanges WERE made, and it will have to run through the sort routines again to see if any more are needed.

What are those "Mysterious Numbers?"

In the Mixed-Text-and-Graphics mode (set by Line 90, remember?), the top twenty lines of "Text" (1-20) are replaced by the graphics display, and the visible portion of the scrolling text window is reduced to the bottom four lines only (21-24).

```
510 VTAB 23
520 PRINT F
525 PRINT CHR$ (7); CHR$ (7)
530 GOTO 200
```

Program Line 510 moves the print cursor to the beginning of the next-to-last (23rd) line in the scrolling window. Line 520 prints the current value of "F" there (which represents the total number of exchanges made during the last pass through the sorting routines).

The automatic carriage return at the end of the print statement puts the print cursor at the beginning of the last line in the scrolling window (#24) and Program Line 525 "prints" a pair of CHR$(7)'s there. I say "prints", because CHR$(7) is really a CONTROL-G, which is not a printable character on the Apple screen. CONTROL-G is the "BELL" command that beeps the Apple's speaker, so what Line 525 does is "ding the bell" twice, and then (here's the real purpose of Line 525!) the automatic carriage return causes the text window to scroll up one line.

This means that the current value of "F" moves up from text line #23 to text line #22, and the previous value of "F" is moved up from text line #22 to text line #21, which is the top of the visible scrolling window. The old top line (which was the PREVIOUS previous value of "F") is bumped off the top and simply disappears. Thus, only two numbers are ever visible at the bottom of the screen.

(Actually, if you watch closely, three numbers are visible for just an instant, during the execution of Line 520 and 525. Then the carriage return at the end of Line 525 scrolls the text and creates the illusion of two changing numbers.)

Line 530 takes us back to the top of the main sort loop, where "F" is reset to zero and we once again step through the blocks, comparing and exchanging as we go.

What Does the Pattern Look Like?

After filling the Low-Res graphics screen with 1600 random-colored blocks (in a 40x40 rectangle), the program sorts the blocks by their numerical color values and arranges them in a rough "fan." The colored "rays" of the fan radiate out from the lower right corner of the rectangle, bulge out near the center, and then taper back in toward the upper left corner. The colors themselves are arranged with the highest numerical value (15 = White) at the lower left side of the fan, and moving diagonally across the fan, up through the colors, to the upper right side of the fan where the lowest numerical color value (0 = Black) is located.

(I must confess, I didn't realize this until the pattern was almost complete! Even after the array of blocks is 3/4 sorted, it still looks pretty disorganized.)

Will it Always Work?

The answer to this one depends on what you mean by "work."

If you take "work" to mean "Will the Bubble Sort algorithm always wind up with each column sorted in ascending order from top-to-bottom, and each row sorted in descending order from left-to-right?," the answer is YES.

If, on the other hand, you define "work" as meaning "Will all the colors be separated into contiguous groups, such that each block of a given color will have at least one neighbor of the same color?," the answer is NO, not always.

The reason why that "perfect" arrangement will not always happen (although it frequently does happen, by chance) is due to the fact that the program is not actually sorting the entire 1600-block array as one group. It is actually sorting eighty 40-block arrays independently. (Forty horizontal rows and forty vertical columns)

Because of the ingenious technique of alternating column sorting with row sorting, the program will *tend* to separate the colors into contiguous groups ... but whether it will succeed or not depends on the initial random distribution of the colors.

(Since the upper-left-hand corner is the last part of the array to be completely sorted, it is there that any "misfits" will likely occur. While I can't provide a proof, I believe there could be a maximum of two isolated blocks.)

What Would You Call it?

Since this program demonstrates a two-dimensional Bubble Sort, using 16 colors, I think "HexaDeca-Color Double-Bubble Demo Sort" tells the tale in a rhythmic way. (Besides, I might get a kick-back from the Fleer's Gum people, who knows?)

A Final Thought ...

If you want to see the color exchanges even more clearly, insert the following line:

```
180 SPEED = 25
```

This will slow the program down to the point where the "bubbling" action of the color exchanges will be very easy to see. The program "picks up" one block and "carries it along" until it bumps into another block of greater value; it will then "drop" the block it's holding and pick up the new block to carry it along for a while. Of course, the program will now take half a day to run ...

```
]LIST

 80  W = 1
 89   HOME
 90   GR
100   FOR I = 0 TO 39
110   FOR J = 0 TO 39
120  C =  INT ( RND (1) * 16)
130   COLOR= C
140   PLOT I,J
150   NEXT J
160   NEXT I
170   GOSUB 1000
200  F = 0
210  V = 38
220  H = 39
300   FOR I = 0 TO H
305   FOR J = 0 TO V
310   IF  SCRN( I,J) =( SCRN( I
      ,J + W) THEN 370
325  G =  SCRN( I,J + 1)
330   COLOR=  SCRN( I,J)
335  F = F + 1
340   PLOT I,J + 1
350   COLOR= G
360   PLOT I,J
370   NEXT J
380   NEXT I
400   FOR I = 0 TO H
405   FOR J = 0 TO V
410   IF  SCRN( J,I)=) SCRN( J
       + W,I) THEN 470
425  G =  SCRN( J + 1,I)
430   COLOR=  SCRN( J,I)
435  F = F + 1
440   PLOT J + 1,I
450   COLOR= G
460   PLOT J,I
470   NEXT J
480   NEXT I
490   GOSUB 1000
500   IF F = 0 THEN  STOP
510   VTAB 23
520   PRINT F
525   PRINT CHR$ (7); CHR$ (7)
530   GOTO 200
1000   REM

 SO WHAT DID YOU EXPECT DEPT

      BY  TOM CRAWFORD

CALL -A.P.P.L.E. * SEPT. 1981

1001   RETURN
```

MONEY BUSINESS

Beat the I.R.S.

You'd always pay the minimum Federal income tax if you had the knowledge and time to examine 120 tax return options. Well, now you can do it, with Datamost's TAX BEATER.

Written by an I.R.S. Enrolled Agent, with 24 years of financial and tax planning experience, the TAX BEATER automatically evaluates up to 120 return options from your input. It searches and finds the best tax path for you . . . displaying up to 15, and ready to print out the optimum method so you pay the minimum!

The TAX BEATER can handle it all. From income averaging to loss carryovers to dividends, alimony, pension to special exclusions. It's perfect for the average taxpayer . . . indispensable for the financial professional. It's so complete, so logical that, especially in this interim year of unusual tax revisions, it can easily pay for itself many times over. Be ready for the I.R.S. . . . with TAX BEATER, the program that's updated for the 1981 tax laws!

$129.95 on disk for Apple II*. Complete with thorough, easy to follow documentation.

Beat the Real Estate game

Real Estate has always been the major wealth-builder. In fact, more millionaire dynasties owe their creation to investments in real estate than anything else. And, it doesn't matter . . . boom times or bust . . . inflation or deflation . . . the knowing investor can make money, even create an empire!

But, the key to real estate success is accurate analysis of opportunities, knowledge of cash flows, return on investment, current and future profitability, tax consequences and other interrelated information — all needed, all necessary to making the right decision.

And that's what REAP (Real Estate Analysis Program) is all about. It takes your input, probes for the right information, scrutinizes the data, the opportunities . . . and automatically delivers the facts . . . showing you the possibilities and why one situation may be superior to another. In short, REAP can help you reap bigger profits. That's why REAP is **the** program for any investor . . . a must program for the sophisticated investor. Don't option, commit, invest or buy without it . . . whether you just want to make good money or create a successful real estate empire!

$129.95 on disk for Apple II*. Complete with thorough easy to follow documentation.

VISA/MASTERCHARGE accepted.
$1.00 shipping/handling charge.
(California residents add 6% tax)

Available at computer stores, or from:

DATAMOST
19273 Kenya St.
Northridge, Ca. 91326
(213) 366-7160

* Apple II is a trademark of Apple Computer, Inc.

CALL —A.P.P.L.E. CONTEST ENTRY COMMENTS

By Luis A. Arana

THE program uses variable W, which is not really necessary, and calls a subroutine that in fact does nothing.

Once I got involved in deciphering what the program was doing, I could not leave these two points unsolved and went a little deeper into sorting techniques. So I learned about another interchange sorting algorithm, called shell sort, remarkably faster than bubble sort, that suggested to me a possible and fairly consistent explanation: the program which is the object of this contest could be an adaptation of another one originally written to implement the shell sort algorithm.

Before going on I need to say something about shell sort. It differs from bubble sort mainly in that the comparison and interchange operations are not limited to neighboring elements. They take place instead between elements separated by a variable increment, which assumes its largest value for the first passes, then is reduced as the sorting progresses. Here seems to lie the key for the speeding up of the process: the time taken by the elements to travel to their final destination can be significantly reduced by allowing them to take larger jumps at first and then smaller jumps as they approach their final position.

And here is my explanation concerning the two unsolved points: W could have been the variable increment and the subroutine at line 1000 responsible for the calculation of its next value after each pass. So, I tried to modify the program looking for an approximate version to what the supposedly original one could have been. Then I introduced the following modifications:

```
80   W = 16
210  V = 39
305  FOR J = 0 TO V-W
325  G = SCRN(I,J + W)
340  PLOT I,J + W
405  FOR J = 0 TO V-W
425  G = SCRN(J + W,I)
440  PLOT J + W,I
490  REM DELETE THIS LINE
500  IF F = 0 THEN GOSUB 1000
1000 IF W = 1 THEN STOP
1001 W = W/2
1002 RETURN
```

After running it, I could verify that shell sort is really faster than bubble sort. The sorting time was about 12 minutes, considerably less than the almost half-hour taken by the bubble version.

This result encouraged me to continue looking for still further improvements on the efficiency of the sorting algorithm. It was an excellent opportunity to put my TED II+ to work and to see for myself how much faster than BASIC, assembly language can be. Then I wrote a subroutine to perform the most time consuming part of the Applesoft program: the nested loops between lines 300 and 480 that actually make up each pass. This time the result could not be more amazing: *the whole array was sorted in less than 18 seconds.*

Listing 1 is the modified BASIC program for working with the assembly language subroutine which is shown as Shell Sort Pass Subroutine 1. It was written with its BASIC counterpart in mind. The corresponding BASIC statements are shown in the assembled listing as comments. This subroutine should be entered and BSAVEd as SUB-1 before running the Applesoft program. Useful instructions to do so can be found in the Reference Manual, page 48, and in the DOS Manual, page 92.

Finally I found that more efficent implementations of shell sort have been developed and I could write a still better version that sorts the array in less than 5 seconds. All the swaps needed for each value of W are performed in only one pass, thus reducing the relation between the total number of comparisons and the amount of swaps effectively carried out. So the time wasted in pure comparison operations, that make no contribution to the blocks movement, is decreased significantly. This version uses only two jump sizes (9 and 1) calculated as suggested by David Weston in his article Comparing Ten Sort Algorithms, *Call* -A.P.P.L.E., Oct. 1980. Listing 2 shows the BASIC portion of this program and Shell Sort Pass Subroutine 2 is the corresponding assembly language subroutine, which should also be entered and BSAVEd as SUB-2 before running the Applesoft program.

For A.P.P.L.E. Members Only

PROGRAM LINE EDITOR

Version 2.0 by Neil Konzen. One of America's top selling utility programs.

CLOSEOUT SPECIAL

$20.00*

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055

Washington State residents add 6.6% sales tax.
Overseas shipping wt. 3 oz.

Watch for announcement of PLE 3.0 coming soon with many new features and improvements. (This will be all new and no update provision is planned)

All foreign orders (incl. Canada) will require additional postage and shipping.

*While supply of documentation lasts.

CALL —A.P.P.L.E. CONTEST COMMENTS

```
]LIST

 10   REM

Shell Sort 1 * Listing 1
     by Luis A. Arana

Call -A.P.P.L.E. : Mar. 1982

 70   PRINT  CHR$ (4)"BLOAD SHELL S
      UB1"
 80 W = 16
 89   HOME
 90   GR
 100   FOR I = 0 TO 39
 110   FOR J = 0 TO 39
 120 C =  INT ( RND (1) * 16)
 130   COLOR= C
 140   PLOT I,J
 150   NEXT J
 160   NEXT I
 170   GOSUB 1000
 200   POKE 0,0: REM
Initialize m.l. variable F (lo)
 210   POKE 1,0: REM
Initialize m.l. variable F (hi)
 220   POKE 5,W: REM
Transfer value of W from BASIC to
      assembly program
 230   CALL 768: REM
Assembly Language subroutine 1
 240 F =  PEEK (0) +  PEEK (1) * 2
      56: REM
Transfer value of F from assembly
      language to BASIC
 500   IF F = 0 THEN  GOSUB 1000
 510   VTAB 23
 520   PRINT F
 525   PRINT  CHR$ (7); CHR$ (7)
 530   GOTO 200
 1000   IF W = 1 THEN  STOP
 1001 W = W / 2
 1002   RETURN
```

```
      ********************************
 2    *                              *
 3    * Shell Sort pass Subroutine 1 *
 4    *        by Luis A. Arana      *
 5    *                              *
 6    * Call -A.P.P.L.E. : Mar. 1982 *
 7    *                              *
      ********************************
 9    *
 10   *
 11   F          EQU   $0
 12   G          EQU   $2
 13   I          EQU   $3
 14   J          EQU   $4
 15   W          EQU   $5
 16   *
 17   PLOT       EQU   $F800
 18   SETCOL     EQU   $F864
 19   SCRN       EQU   $F871
 20   *
 21   *
 22              ORG   $300
 23   *
 24   *
 25   *
 26   * Starting column sweeps
 27   *
```

```
]
]LIST

 10   REM

Shell Sort 2 * Listing 2
     by Luis A. Arana

Call -A.P.P.L.E. : Mar. 1982

 60   PRINT  CHR$ (4)"BLOAD SHELL S
      UB2"
 70 D = 3
 80 E(2) = 9:E(1) = 1
 89   HOME
 90   GR
 100   FOR I = 0 TO 39
 110   FOR J = 0 TO 39
 120 C =  INT ( RND (1) * 16)
 130   COLOR= C
 140   PLOT I,J
 150   NEXT J
 160   NEXT I
 170   GOSUB 1000
 200   POKE 0,0: REM
Initialize m.l. variable F (lo)
 210   POKE 1,0: REM
Initialize m.l. variable F (hi)
 300   POKE 6,W: REM
Transfer value of W from BASIC to
      assembly program
 320   CALL 768: REM
Assembly Language subroutine 2
(Column sweeps)
 325   CALL 858: REM
Assembly Language subroutine 2
(Row sweeps)
 330 F =  PEEK (0) +  PEEK (1) * 2
      56: REM
Transfer value of F from assembly
      language to BASIC
 500   VTAB 23
 510   PRINT F
 520   GOSUB 1000
 525   PRINT  CHR$ (7); CHR$ (7)
 530   GOTO 200
 1000   IF D = 1 THEN  STOP
 1001 D = D - 1
 1002 W = E(D)
 1003   RETURN
```


DOUBLE your DISKETTES

The only reasons your APPLE II cannot use the back side of your diskette are:

1. There is no notch.
2. The diskette manufacturer either did not test the back side, or worse, put the flawed side to the back.

A pair of diagonal cutters will solve problem number 1.

DISK PREP

will solve problem number 2.

DISK PREP formats and tests your disk. Sectors with flaws are left so they cannot be used. Your disk is left ready to boot, complete with a flaw report program saved on it.

$25.00

sympathetic software

9531 Telhan Drive
Huntington Beach, Ca. 92646

DOS 3.3 and DOS 3.2.1
California residents add $1.50 tax.

```
0300: A9 00      28              LDA  #0          ;I=0
0302: 85 03      29              STA  I
0304: A9 00      30  ILOOP       LDA  #0          ;J=0
0306: 85 04      31              STA  J
                 32  *
                 33  * Individual column sweep
                 34  *
0308: A5 04      35  JLOOP       LDA  J           ;G=SCRN(I,J+W)
030A: 18         36              CLC
030B: 65 05      37              ADC  W
030D: A4 03      38              LDY  I
030F: 20 71 F8   39              JSR  SCRN
0312: 85 02      40              STA  G
0314: A5 04      41              LDA  J           ;IF SCRN(I,J) <= SCRN(I,J+W)
0316: 20 71 F8   42              JSR  SCRN
0319: C5 02      43              CMP  G
031B: 30 1D      44              BMI  NEXTJ       ;THEN GOTO NEXTJ
031D: F0 1B      45              BEQ  NEXTJ
031F: 20 64 F8   46              JSR  SETCOL      ;COLOR=SCRN(I,J)
0322: E6 00      47              INC  F           ;F = F+1
0324: D0 02      48              BNE  P1
0326: E6 01      49              INC  F+1
0328: A5 04      50  P1          LDA  J           ;PLOT I,J+W
032A: 18         51              CLC
032B: 65 05      52              ADC  W
032D: 20 00 F8   53              JSR  PLOT
0330: A5 02      54              LDA  G           ;COLOR=G
0332: 20 64 F8   55              JSR  SETCOL
0335: A5 04      56              LDA  J           ;PLOT I,J
0337: 20 00 F8   57              JSR  PLOT
033A: E6 04      58  NEXTJ       INC  J           ;J = J+1
033C: A9 27      59              LDA  #39         ;IF J <= 39-W
033E: 38         60              SEC
033F: E5 05      61              SBC  W
0341: C5 04      62              CMP  J
0343: 10 C3      63              BPL  JLOOP       ;THEN GOTO JLOOP
0345: E6 03      64  NEXTI       INC  I           ;I = I+1
0347: A9 27      65              LDA  #39         ;IF I <= 39
0349: C5 03      66              CMP  I
034B: 10 B7      67              BPL  ILOOP       ;THEN GOTO ILOOP
                 68  *
                 69  * Starting row sweeps
                 70  *
034D: A9 00      71              LDA  #0          ;I=0
034F: 85 03      72              STA  I
0351: A9 00      73  ILOOP2      LDA  #0          ;J=0
0353: 85 04      74              STA  J
                 75  *
                 76  * Individual row sweep
                 77  *
0355: A5 04      78  JLOOP2      LDA  J           ;G=SCRN(J+W,I)
0357: 18         79              CLC
0358: 65 05      80              ADC  W
035A: A8         81              TAY
035B: A5 03      82              LDA  I
035D: 20 71 F8   83              JSR  SCRN
0360: 85 02      84              STA  G
0362: A5 03      85              LDA  I           ;IF SCRN(J,I) >= SCRN(J+W,I)
0364: A4 04      86              LDY  J
0366: 20 71 F8   87              JSR  SCRN
0369: C5 02      88              CMP  G
036B: 10 20      89              BPL  NEXTJ2      ;THEN GOTO NEXT J2
036D: 20 64 F8   90              JSR  SETCOL      ;COLOR = SCRN(J,I)
0370: E6 00      91              INC  F           ;F = F+1
0372: D0 02      92              BNE  P2
0374: E6 01      93              INC  F+1
0376: A5 04      94  P2          LDA  J
0378: 18         95              CLC
0379: 65 05      96              ADC  W
037B: A8         97              TAY
037C: A5 03      98              LDA  I           ;PLOT J+W,I
037E: 20 00 F8   99              JSR  PLOT
0381: A5 02     100              LDA  G           ;COLOR = G
0383: 20 64 F8  101              JSR  SETCOL
0386: A5 03     102              LDA  I           ;PLOT J,I
```

Be a Part of Our Success

SDS MOVES AHEAD

We at SDS have built our reputation by offering the highest quality utility software available, combined with personalized service and after-sale technical support. Our total involvement in the utility software area is further reflected in our president Roger Wagner's widely read monthly column, "Assembly Lines". Our thorough knowledge of the field, and our sense of commitment to all our customers have led to tremendous success in 1981.

So SDS is growing - from the creative vision of a single person, to one of the shining stars of the software industry. Whether you're an end user, dealer, or author, we want you to make SDS part of your success with the Apple II.

To make that happen, we're now offering a wider range of utility and other software, more customer service, better author benefits. The service is still personal and the products are still the finest anywhere - but now we can offer you more . . .

SDS southwestern data systems™

MORE NEW PROGRAMS

ROUTINE MACHINE – A terrific way to put machine language in your Applesoft programs. Simply choose from a pre-written library of subroutines. No knowledge of machine language is required.

AMPERSOFT PROGRAM LIBRARY – A set of additional machine language subroutines for use with Routine Machine.

MERLIN – A powerful, full-featured 6502 macro assembler.

ASCII EXPRESS: THE PROFESSIONAL A professional version of our best selling communication package that transfers files of ANY size or type. Supports all 80 column boards, interactive macros, and much more.

P-TERM: THE PROFESSIONAL – A Pascal version of Bill Blue's highly acclaimed communication software.

DOUBLETIME PRINTER – Increase your Apple's productivity by simultaneously printing text or binary files from one program while you run other programs.

UNIVERSAL GRAPHICS – The ultimate Hi-Res printing utility to transform your creations into permanent records on ANY graphic printer. Includes picture cropping, inverse inking, horizontal or vertical printouts, and a range of magnifications.

MORE DEALER SUPPORT

SDS is committed to the continuing support of the Apple dealer. As part of our new program, SDS offers:

- Informative point-of-sale materials
- Comprehensive national advertising
- Head Start Program to insure priority delivery of all new products
- Special dealer discount on in-store display items
- Comprehensive line of wide application software products
- Full-time technical support staff

SDS
southwestern
data systems™

P.O. Box 582C Santee, CA 92071
Tel. 714-562-3670

MORE AUTHOR BENEFITS

SDS is an industry pioneer in innovative software for the Apple II. When you join the SDS team, you get the benefits of a professional support staff experienced in providing all the help you need to market your program. Other benefits include:

- Technical programming assistance
- Unique copy protection with limited backups
- Successful marketing strategies
- Assistance in writing the manual
- Professional product artwork
- Quality advertising
- Superior packaging
- National distribution
- Highest royalties paid monthly
- Customer service support

Here's the chance for you to market your program with the finest publisher in the software industry. Let Southwestern Data Systems' reputation and proven track record go to work for you. Be a part of our success – call today.

```
0388: A4 04                  LDY  J
038A: 20 00 F8               JSR  PLOT
038D: E6 04        NEXTJ2    INC  J          ;J = J+1
038F: A9 27                  LDA  #39        IF J <= 39-W
0391: 38                     SEC
0392: E5 05                  SBC  W
0394: C5 04                  CMP  J
0396: 10 BD                  BPL  JLOOP2     ;THEN GOTO JLOOP2
0398: E6 03        NEXTI2    INC  I          ;I = I+1
039A: A9 27                  LDA  #39        ;IF I <= 39
039C: C5 03                  CMP  I
039E: 10 B1                  BPL  ILOOP2     ;THEN GOTO ILOOP2
03A0: 60                     RTS

--END ASSEMBLY--

ERRORS: 0

161 BYTES
```

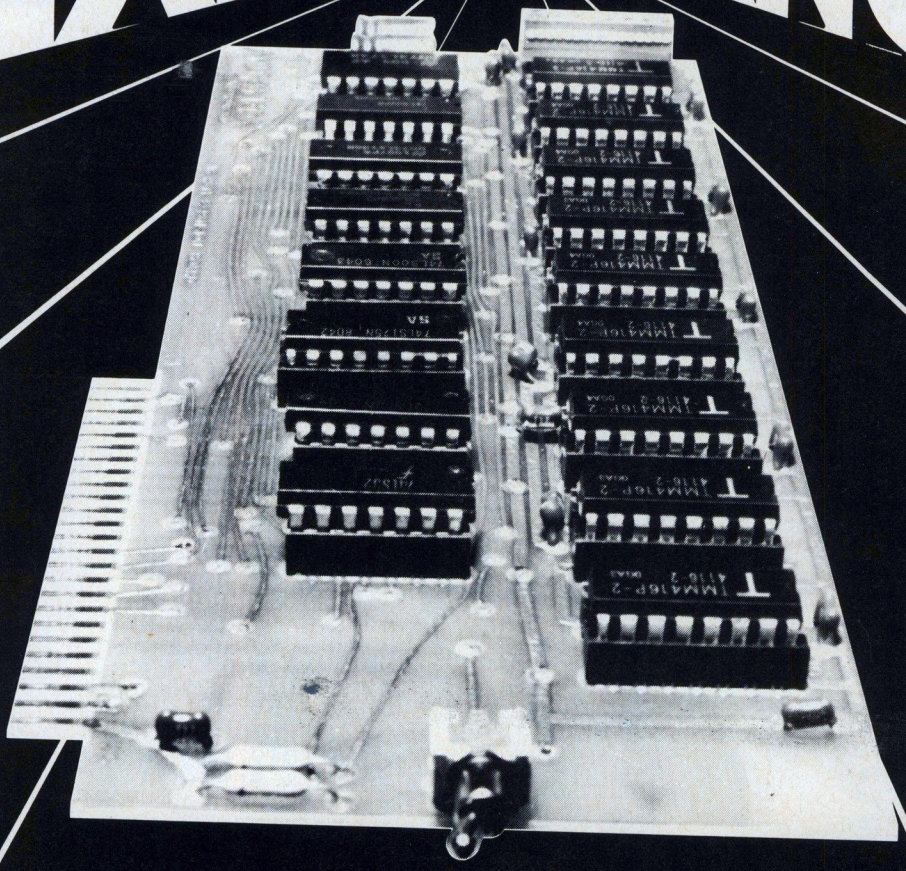
```
:ASM
                    ********************************
                    *                              *
                    * Shell Sort pass Subroutine 2 *
                    *       by Luis A. Arana       *
                    *                              *
                    * Call -A.P.P.L.E. : Mar. 1982 *
                    *                              *
                    ********************************
                    *
                    *
                    F         EQU  $0
                    G         EQU  $2
                    I         EQU  $3
                    J         EQU  $4
                    K         EQU  $5
                    W         EQU  $6
                    *
                    PLOT      EQU  $F800
                    SETCOL    EQU  $F864
                    SCRN      EQU  $F871
                    *
                    *
                              ORG  $300
                    *
                    *
                    *
                    * Starting column sweeps
                    *
0300: A9 00                   LDA  #0         ;I=0
0302: 85 03                   STA  I
0304: A5 06         ILOOP     LDA  W          ;J=W
0306: 85 04                   STA  J
                    *
                    * Individual column sweep
                    *
0308: A5 04         JLOOP     LDA  J          ;G=SCRN(I,J+W)
030A: A4 03                   LDY  I
030C: 20 71 F8                JSR  SCRN
030F: 85 02                   STA  G
0311: A5 04                   LDA  J          ;K = J-W
0313: 38                      SEC
0314: E5 06                   SBC  W
0316: 85 05                   STA  K
0318: A5 05         KLOOP     LDA  K          ;IF G >= SCRN(I,K)
031A: A4 03                   LDY  I
031C: 20 71 F8                JSR  SCRN
031F: C5 02                   CMP  G
0321: 30 15                   BMI  LCHNG      ;THEN GOTO LCHNG
0323: 20 64 F8                JSR  SETCOL     ;COLOR = F+1
0326: 20 4F 03                JSR  NPLOT      ;PLOT I,K+W
0329: E6 00                   INC  F          ;F = F+1
032B: D0 02                   BNE  NEXTK
032D: E6 01                   INC  F+1
```

```
032F:  38           54  NEXTK   SEC             ;K = K-W
0330:  A5 05        55          LDA   K
0332:  E5 06        56          SBC   W
0334:  85 05        57          STA   K
0336:  10 E0        58          BPL   KLOOP     ;IF K >= 0 THEN GOTO KLOOP
0338:  A5 02        59  LCHNG   LDA   G         ;COLOR = G
033A:  20 64 F8     60          JSR   SETCOL
033D:  20 4F 03     61          JSR   NPLOT     ;PLOT I,K+W
0340:  E6 04        62          INC   J         ;J = J+1
0342:  A9 27        63          LDA   #39       ;IF J <= 39
0344:  C5 04        64          CMP   J
0346:  10 C0        65          BPL   JLOOP     ;THEN GOTO JLOOP
0348:  E6 03        66          INC   I         ;I = I+1
034A:  C5 03        67          CMP   I         ;IF I <= 39
034C:  10 B6        68          BPL   ILOOP     ;THEN GOTO ILOOP
034E:  60           69          RTS
034F:  A5 05        70  NPLOT   LDA   K         ;PLOT I,K+W
0351:  18           71          CLC
0352:  65 06        72          ADC   W
0354:  A4 03        73          LDY   I
0356:  20 00 F8     74          JSR   PLOT
0359:  60           75          RTS
                    76  *
                    77  * Starting row sweeps
                    78  *
035A:  A9 00        79          LDA   #0        ;I=0
035C:  85 03        80          STA   I
035E:  A5 06        81  ILOOPH  LDA   W         ;J=W
0360:  85 04        82          STA   J
                    83  *
                    84  * Individual row sweep
                    85  *
0362:  A5 03        86  JLOOPH  LDA   I         ;G = SCRN(J,I)
0364:  A4 04        87          LDY   J
0366:  20 71 F8     88          JSR   SCRN
0369:  85 02        89          STA   G
036B:  A5 04        90          LDA   J         ;K = J-W
036D:  38           91          SEC
036E:  E5 06        92          SBC   W
0370:  85 05        93          STA   K
0372:  A5 03        94  KLOOPH  LDA   I         ;IF G <= SCRN(K,I)
0374:  A4 05        95          LDY   K
0376:  20 71 F8     96          JSR   SCRN
0379:  C5 02        97          CMP   G
037B:  B0 15        98          BCS   LCHNGH    ;THEN GOTO LCHNGH
037D:  20 64 F8     99          JSR   SETCOL    ;COLOR = SCRN(K,I)
0380:  20 A9 03    100          JSR   NPLOTH    ;PLOT K+W,I
0383:  E6 00       101          INC   F         ;F = F+1
0385:  D0 02       102          BNE   NEXTKH
0387:  E6 01       103          INC   F+1
0389:  38          104  NEXTKH  SEC             ;K = K-W
038A:  A5 05       105          LDA   K
038C:  E5 06       106          SBC   W
038E:  85 05       107          STA   K
0390:  10 E0       108          BPL   KLOOPH    ;IF K >=0 THEN GOTO KLOOPH
0392:  A5 02       109  LCHNGH  LDA   G         ;COLOR = G
0394:  20 64 F8    110          JSR   SETCOL
0397:  20 A9 03    111          JSR   NPLOTH    ;PLOT K+W,I
039A:  E6 04       112          INC   J         ;J = J+1
039C:  A9 27       113          LDA   #39       ;IF J <=39
039E:  C5 04       114          CMP   J
03A0:  10 C0       115          BPL   JLOOPH    ;THEN GOTO JLOOPH
03A2:  E6 03       116          INC   I         ;I = I+1
03A4:  C5 03       117          CMP   I         ;IF I <=39
03A6:  10 B6       118          BPL   ILOOPH    ;THEN GOTO ILOOPH
03A8:  60          119          RTS
03A9:  A5 05       120  NPLOTH  LDA   K         ;PLOT I,K+W
03AB:  18          121          CLC
03AC:  65 06       122          ADC   W
03AE:  A8          123          TAY
03AF:  A5 03       124          LDA   I
03B1:  20 00 F8    125          JSR   PLOT
03B4:  60          126          RTS

--END ASSEMBLY--      ERRORS: 0     181 BYTES
```

NEW LOW PRICE
DIRECT FROM MANUFACTURER
$120.00

EXPAND ☆ ENHANCE



16K RAM EXPANSION BOARD FOR THE APPLE II* $120.00

The Andromeda 16K RAM Expansion Board allows your Apple to use RAM memory in place of the BASIC Language ROMs giving you up to 64K of programmable memory. Separate Applesoft* or Integer BASIC ROM cards are no longer needed. The 16K RAM Expansion Board works with the Microsoft Z-80 card, Visicalc, DOS 3-3, Pascal, Fortran, Pilot, and other software. A switch on the card selects either the RAM language or the mainboard ROMs when you reset your Apple.

The Andromeda 16K RAM Expansion Board has a proven record for reliability with thousands of satisfied customers.

Now with One Year Warranty.

ANDROMEDA
☆ INCORPORATED

Greensboro, NC. 27410
P.O. Box 19144

919 852-1482

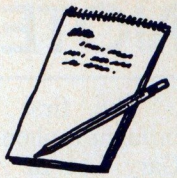
VISA

master charge
THE INTERBANK CARD

Price for Andromeda 16K RAM expansion board now only $120.00. Please add $5 for shipping and handling. North Carolina residents add 4% sales tax.

*DEALER INQUIRIES WELCOME.

NOTES FROM A NOVICE



By Charles W. Rusch/Call —A.P.P.L.E. Staff Writer

THIS is the third "Notes from a Novice." The idea is to have *Notes* appear as a regular column every other month, but we've had some start-up difficulties finding room. So the first collection of *Notes* was written in June and published in September, the second was written in September and published in January, and this one is being written in January to appear in March. We have consequently had some trouble fulfilling another of our intentions, which is to make it a column which is responsive to reader input.

In this issue, I'll be responding to several letters and phone calls I received on the first *Notes,* and finishing up all the items I started last time. Then I'll go on to a little more of the disk doctoring we started in the first column, and also a few of the more interesting tips, tricks, and news items I've run into since then – for example, my experiences with the new keyboard Enhancer II from Videx.

But first, the feedback (I hope no one who wrote me in September has been sitting by the mailbox). The first response I got to Notes #1 was from a friend of mine who asked how long I thought I could pass myself off as a novice. I modestly responded, "I hope forever" – and I meant it; I love learning. When I'm learning something new, I'm excited, alive, and full of energy. I think that is why I got hooked on microcomputers – they are great learning machines. As a result, as soon as I get one thing "down," I go on to another, so it is clear I shall be a novice for a very long time indeed. I will admit that *mastery* is quite another thing, and comes to me more slowly, to put it gently.

The day the September issue arrived, I got a call from someone somewhere in the country, who just wanted to tell me how great the article was. I was so stunned I didn't know what to say, so we started to chat – and then the coins dropped. Apparently, whoever it was called from a phone booth (!), and since there was no point in calling back to say good-bye, it was suddenly over. Whoever you are out there, your call definitely made my day. In the future, if any of you call long distance, I'll try not to gab.

Another call came from someone at Apple Computer telling me where I could find instructions on CTRL-S in the Apple manuals, which I appreciated. I'm embarrassed to say CTRL-S appears in every manual at least once, and many times in the Applesoft Tutorial. But nobody can know everything!

A friend of mine, who got his Apple about the time I did, told me it was six months before he learned about clearing the screen with the HOME command. I had to tell another friend, who is an excellent Pascal programmer, that you can reboot another disk by pressing "H" from command level. We all suffer our individual oversights.

Just as I was writing this, I got a call from a Matt Berman, in New Orleans, with an easy way to make sure that all lines to be printed on the screen between quotes are 40 characters long with no words split at the ends. You know the scene, every line has to have 40 characters, but you're not sure when the next line should begin, so you insert a bunch of spaces to try to keep words from being split, and then you run the program to see how you did. Or else you get a lot of practice counting up to 40. After a couple of years messing around like this, he realized that after he typed

```
PRINT "
```

he could use the escape cursor moves to drop down to the beginning of the next line and start it at the left margin. Ditto for successive lines. Pretty simple.

I've found that using PLE (Program Line Editor by Neil Konzen), the lines are a full 40 characters and I can just type each line until it wraps around to its beginning again. I think I like Matt's method better, because it shows you how the lines will look on the screen when run – that is, until you list the program, and Applesoft messes it up again.

After reading my article, one of the local store managers told me that it really is not too difficult to replace power switches, and that he had done several. However, the best comment on the subject was from L. Garrett Sliger, in Lake Forest, California, who was also having trouble with his power switch and replaced it himself. He says that the old switch can be pried out of the power supply housing with a small screwdriver. Once that is done, "you buy a replacement switch for a dollar and a dime, solder the wires to the correct terminal using the old one as a model, and simply push the new switch back in the power supply."

I haven't tried it yet, but it's tops on my *to do* list. I like that kind of letter.

I got another excellent letter from a Dr. Thomas E. Militello, MD, who liked the column enough to join *A.P.P.L.E.,* and asked for advice on getting started using Pascal. Apparently, he already has the Language System (probably for reasons other than learning Pascal), but has never used Pascal. I shall try to respond to Dr. Militello in the next column (which will be on learning a new language), although I am sure he is well on the road to Pascal wizardry by now.

In the meantime, I've heard that the book, *Apple Pascal: A Hands On Approach* is an excellent tutorial. I am most of the way through *Introduction to Pascal,* by Rodney Zaks, and find it very clear and helpful. Currently, I am teaching myself both Pascal and TransFORTH II (how's that for a combination?).

Speaking of books, don't miss *The Soul of a New Machine,* by Tracy Kidder. It's a behind-the-scenes account of the design of Data General's *Eagle* mini-computer, and reads like a mystery thriller with lots of insights into computer design, manufacturing and marketing.

NOTES FROM A NOVICE

The first *Notes* column also brought a phone call and letter from Gary Lee of *Accent Software,* who said that everything I said about debugging Applesoft programs can now be done much easier with the aid of their SOFT-STEP program. If I ever get back to Applesoft programming, I'll try it out.

Beginning response on Notes #2, about customizing your Apple PIE keyboard, produced a call from Jim DeWitt, in Alaska, who wanted to know how to get an 80 column version of Apple PIE now that Programma International has folded. What I understand is that Hayden Software, who took over Programma, is releasing a new improved version of Apple PIE which they are calling *PIE Writer.* So that is one way to go.

Unfortunately, I hear it is copy-protected. One of the best features of the old PIE is that it is *not* protected, allowing the production of standard DOS binary and text files, as well as its modification to suit your own needs, out of which fact came my last column on customizing your Apple PIE keyboard. As for getting an official update, better check with Hayden to find out whether they are going to continue to support the old version as well as the new. If not, I am afraid that Hayden is forcing the old PIE to go underground by protecting the new one.

The only other question I have received (as of this writing) about customizing your PIE keyboard was whether the procedure I gave is also true for the 80 column version. The answer is yes. The control character tables in both versions are exactly the same, as are their addresses. I have modified both the 40 and 80 column versions with those procedures.

Hardware Tips

The war between software fixes and hardware fixes for the Apple II goes on.....Everything I told you about customizing your PIE keyboard can now be done much more quickly and easily through hardware. Furthermore, it can be done for all of your programs, not just PIE.

What I am saying is that I am now the proud owner of the new keyboard enhancer from Videx, Inc. called Enhancer II. It does everything the old one did, like giving you a full ASCII upper and lower case character set, true shift and shift/lock, etc., but the big news is it also has programmable keys!

With it you can create up to 512 bytes of your own macros (single key strokes which produce whole words or commands). This means you can reconfigure the entire keyboard, *including* throwing out the QWERTY arrangement of the keyboard characters for something wonderfully efficient like the Dvorak keyboard.

Want to learn to type all over again? It'll be faster this time... The secret to all these marvels is that Enhancer II has its own on-board microprocessor! Like the Keyboard and Display Enhancer (K.A.D.E.), it tucks under the keyboard, but unlike K.A.D.E., it replaces the encoder board, making the installation elegant and easy.

The Enhancer II comes with a utilities diskette and with several sets of macros all ready to download and a number of nifty utilities programs, including a fine macro editor. The real fun for me so far has been programming my own control keys for the various programs I use. I now think it might be possible to design *one* set of my favorite control key functions (cursor moves, search and replace, etc.), and have those automatically downloaded into the keyboard to convert every program I use to my standard set.

So far, I have only partially reached that goal. Because the macros are read first, my macro set of control keys has precedence over those which came with the program, and there is no need to mess around inside the program code, as we did in the January column.

With the Enhancer II, you can reprogram every character key on your keyboard, any control key, or any control/shift key, and in two modes, CAPS LOCK and upper/lower case. That gives a lot of choices. And my experience so far would indicate that 512 bytes for each set of macros is plenty. Every set can be stored on disk as a binary file and simply BRUN to download it into the keyboard, where it will remain until you power down or load another set into the keyboard on top of it. The most convenient arrangement is to have the HELLO program on each disk BRUN the appropriate macro set into the keyboard automatically when the disk is booted. I haven't had the nerve to try out the Dvorak arrangement yet, but Enhancer II sures does a job on old QWERTY.

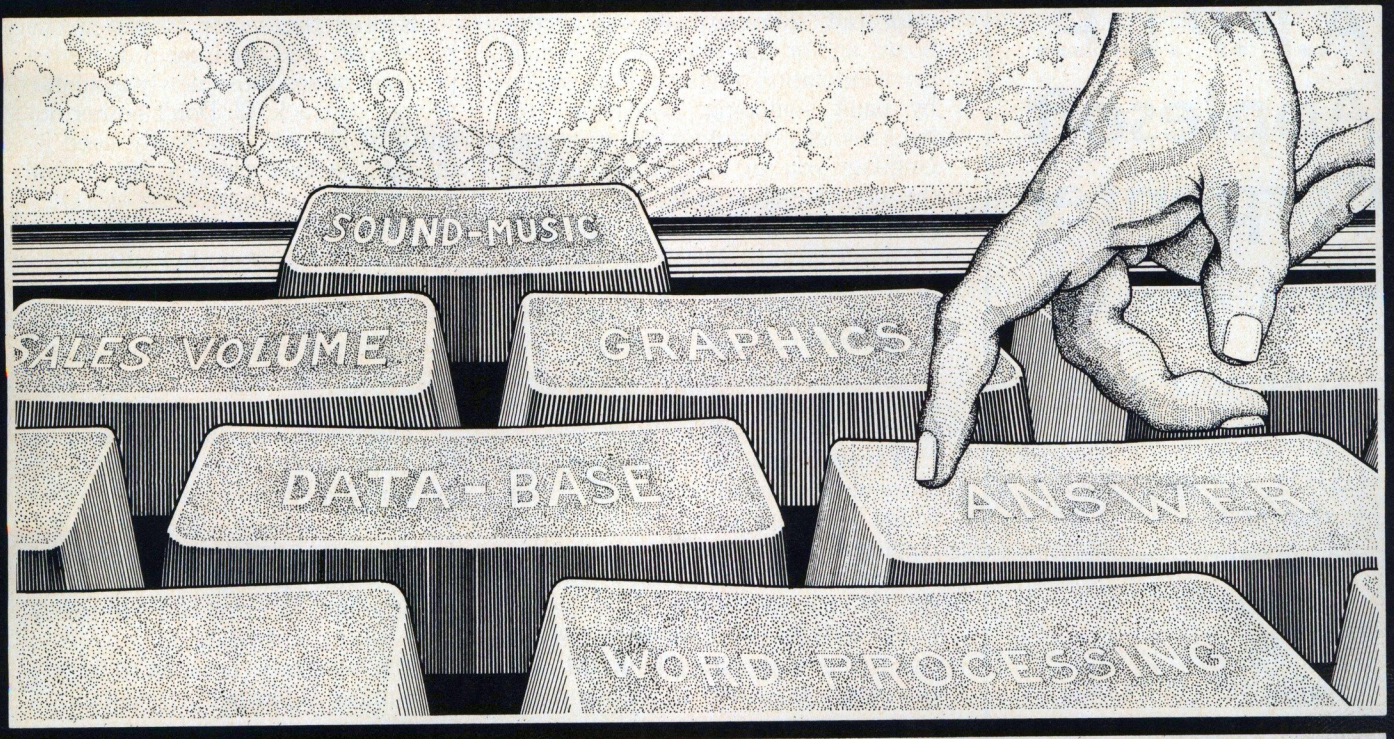
Among the software utilities on the Enhancer II disk, is a program by Ron Aldrich called QUICK LOADER, which you can insert into the front end of any binary file and it will cut the load time to about a third of its original length (even better on long programs). The Enhancer II utilities disk itself loads DOS *and* one of the BASIC languages into the language card (if found - not required) in about six seconds. I put QUICK LOADER on the front of my Apple PIE program and it now loads in 5 seconds instead of 14. FORMAT loads in 4 seconds. So the wait time to switch back and forth between those two programs is now very tolerable.

I haven't been this excited by a piece of hardware in months. I don't think there is anything like it on the market, not even among Lazer's extensive keyboard collection (I am ready to stand corrected).

New subject - large memory boards are springing up all over the place. First there was Saturn System's 32k Ram Board, then the 64k Legend board, now the 256k APP-L-CACHE from Sorrento Valley Associates (SVA). By the time this is printed, I am sure there will be more.

There are in fact several more, from 64 to 128 to 256k, and more coming. . . .ed.

THE PROGRAMMER™ HAS THE ANSWERS TO YOUR CUSTOM SOFTWARE QUESTIONS.



Mass produced software has rapidly approached levels of sophistication beyond the imagination of a year ago. However, regardless of the sophistication of this software, every user still encounters the situation where he thinks to himself "If it only did this or had that capability . . ." Now, IBM® has recently brought the power and versatility of their main-frame systems to your fingertips with the new IBM personal microcomputer, and now Advanced Operating Systems complements this accomplishment by introducing THE PROGRAMMER, a software package which brings you abilities previously possessed only by a highly skilled computer programmer.

WHAT DO YOU HAVE TO KNOW?

You must have a clear idea of what you want your program to do. THE PROGRAMMER can't read your mind, but, with your direction, it can actually write the lines of BASIC language necessary for the micro-computer to perform your tasks. *Non-programmers* with little or no knowledge of BASIC-language programming can use the simple, multiple-choice, menu format to create all types of programs.

WHAT DO YOU HAVE TO DO?

You direct THE PROGRAMMER via a series of "menus". Each menu presents a question and a list of the available choices. You simply input the number of the function you want, and THE PROGRAMMER writes the corresponding BASIC program lines. The finished product is automatically stored on the disk, from which it can be copied on another disk for later use.

WHAT CAN THE PROGRAMMER DO FOR YOU?

THE PROGRAMMER can write a _____ (choose the correct response.)

a. data-base program to develop and update a mailing list or keep track of catalog items in your inventory.
b. graphics generator program.
c. program to create sounds or music.
d. customized small business accounting system.
e. program to interface with another computer device.
f. word processing program to print department reports.
g. all of the above, and more.

The correct response to this sample menu is "g". The "bottom line" is that THE PROGRAMMER will write a program for any purpose. The possibilities are limited only by your imagination. Once a program is completed it can be saved on a disk, allowing you to write additional programs. THE PROGRAMMER will not create the ideas, but will permit you to quickly and easily implement any programming ideas you have.

HOW, WHEN, AND WHERE CAN YOU GET IT?

THE PROGRAMMER is not a fanciful dream for the future, but is finished and available at your local software retailer, or call (800) 348-8558 to order. (Indiana residents, call (219) 879-4693.) MasterCard and VISA accepted.

THE PROGRAMMER for the IBM® Personal Computer $495

Available soon for:
APPLE II® or APPLE II Plus®, 48K
APPLE with CP/M® Card, 64K

ADVANCED OPERATING SYSTEMS

450 St. John Road
Michigan City, IN 46360

*IBM is a registered trademark of International Business Machines.
*APPLE II, APPLE II Plus are registered trademarks of Apple Computer, Inc.
*CP/M is a registered trademark of Digital Research, Inc.

A.P.P.L.E. members only!

The A.P.P.L.E. Applesoft Carpenter

by Wayne Eastwood

AVAILABLE NOW

- LISTER: a program list formatter
- COMPARER: compares two programs for differences
- CRUNCHER: compacts programs, strips REMs
- CONVERTER: converts Integer to Applesoft, corrects all syntax — Integer Basic not required in system.
- FINDER: locates any string, token or variable and lists lines
- KEEPER: saves variables while editing program lines

Fast machine language routines

$18.50

Overseas shipping weight: 2.5 oz.

A.P.P.L.E. Orders
304 Main Ave. S. Suite 300
Renton, WA 98055
(206) 271-4514
Visa, M/C welcome

All foreign orders (including Canada) will require additional postage and shipping.

NOTES FROM A NOVICE

Each package includes a disk of software allowing DOS to be relocated automatically onto the board and/or the selection of a disk drive emulation program. It all works by bank switching, with one 16k bank usually reserved for whatever language you are loading into it, like the alternate form of BASIC, Pascal, FORTRAN, or whatever. I have only tried the Saturn board so far (and have only seen the Legend demonstrated), but the 32k board was apparently carefully switched and user-transparent (meaning we novices wouldn't even know the board is there).

I suspect that we'll be seeing a lot more of this sort of peripheral. As an added bonus, the user's old Integer or Applesoft firmware card, and/or Language Card can be used in another slot by changing a few addresses in DOS. So then you have both the old and the new monitors in ROM and even more memory. To an assembly language type that makes a difference.

Quickies of Possible Interest

The June '81 issue of *Call –A.P.P.L.E.* had a letter by Dr. Warren Ostlund saying that you could get 13 extra sectors of disk space back from DOS, which doesn't use it. Since then, I've read in two different places that it's not a good idea to try to store data on track 00. (If you would like a partial explanation which I do not fully understand yet, see page 4-6 in *Beneath Apple DOS).* But that was only two of the Doctor's free sectors anyway, and it still leaves 11 on track 02 which might come in handy in a pinch. But how to do it?

What's involved is changing bytes 40 and 41 in VTOC, DOS's Table of Contents, which is on track 17, sector 00 (see September's *Notes).* To write to that sector you'll need a program like ZAP (Beneath Apple DOS), or DISKZAP, on *A.P.P.L.E.'s* Diskpak 6B-16. If you use the latter, the track to be modified will be stored in the 9600 area of memory. So when you get through all the preliminaries, and are ready to write to the disk, just enter

*** 9640:FF E0**

Reread the sector to see if the patch took, and then use a disk map program to see those newly available sectors on track 02. Don't bother doing it at all until you really need it, though, because using a "fix sector count," or a "reconstruct VTOC" program, will mark those sectors as full again before you get to use them.

A more sophisticated and involved method, which also wipes out DOS, but gives you up to 70 additional data sectors may be found in Clay Ruth's story in this issue.

There's an excellent Glossary in the back of *Beneath Apple DOS.* It includes definitions of such mysteries as ASCII, RWTS, T/S lists, and VTOC. Two I puzzled over for some months which are not in it, were *baud rate* (the butt of many bad and baudy body jokes), and *OEM.*

The baud rate turns out to be just the number of bits per second which a communications device like a modem or a plotter will transmit. OEM was much tougher. It kept appearing in ads. One ad even had a box for me to check indicating whether I was an OEM or a "dealer." I wasn't even sure what kind of *dealer* they had in mind; how was I to know if I was an OEM? What is an OEM? An Obviously Emasculated Male? (Me? Hard to tell; I'll ask my wife...). A One-Eyed Maladroit? (Well, hardly...). An Ominously Eager Manufacturer? (Not me, but I felt I was getting close...). An Overly Enthusiastic Manufacturer? (Very close...).

Then one day, after about six months watching to see if anyone would ever define OEM, there it was in Sol Libes' "Bytelines" column in the August issue of *Byte:* Original Equipment Manufacturer! Suddenly, the sky parted, and I thought I heard the sound of CTRL-G's in the distance.....

Send me your favorite mystery words.

See you later.....

ECHO][™ SPEECH SYNTHESIZER

UNLIMITED VOCABULARY AT YOUR FINGERTIPS

Give your Apple* something to talk about with an ECHO][Speech Synthesizer. The ECHO][offers intelligible voice-output while using a minimum of RAM. The ECHO][uses LPC technology pioneered by Texas Instruments, coupled with a phoneme-based operating system allowing you to create any vocabulary desired. Variable stress, pitch and volume let your Apple ask questions or make exclamations while also allowing for optimal quality. Speech can easily be added to BASIC programs with PRINT statements.

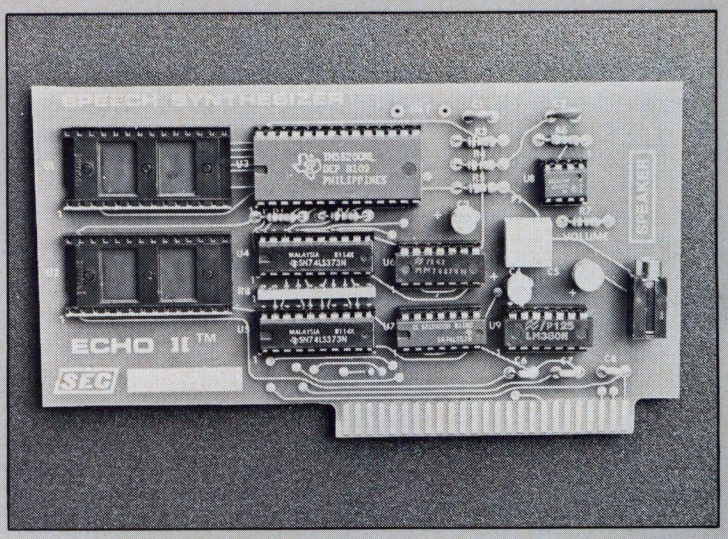
The applications of synthesized speech are limited only by your imagination. From educational programs to games to business applications, speech-output adds a new dimension to your Apple's capabilities.

The ECHO][Speech Synthesizer comes complete with speaker and cable, instruction manual, speech editing features and a sample vocabulary. The ECHO][requires 48K, Applesoft and at least one disk drive. Suggested list price is $225.

For further information, contact your dealer or Street Electronics Corporation.

Dealer inquiries welcome.

*** Trademark of Apple Computer Company**



SEC **STREET ELECTRONICS CORPORATION**

3152 E. La Palma Ave., Suite D
Anaheim, CA 92806 **(714) 632-9950**

"looking for ze card?"

On close inspection "ze card" is really Zi-GUARD . . . an honest-to-goodness "private eye" of your own. If you're an Apple owner.

The only total burglar and fire detection alarm system available for your Apple, Zi-GUARD is introduced by Zena micro engineering.

Give it ze try!

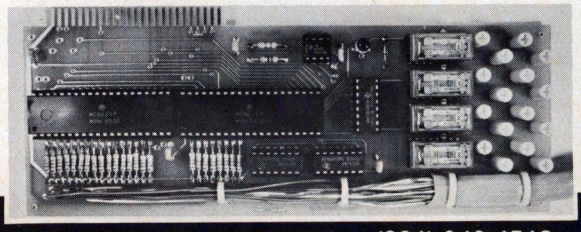
SOFTWARE
Complete program package, including Hi-Res graphic floor-plan displays. Full color and sound.

BACKGROUND
Program runs independently in background, unnoticed until needed. Internal Real-Time clock.

POWERFUL
Eight output options. Up to four relays ON BOARD, **plus** BSR interface port.

FLEXIBLE
All options are user-definable. Twenty-four separate inputs for intrusion sensors and fire detectors.

Dealer Inquiries Welcome



SYSTEM ONE
Zi Guard Security Card, Cable and connectors, software on diskette, one DPDT relay, instruction manual.

SPECIAL INTRODUCTORY OFFER $149

SYSTEM TWO
For a total package, get all of the above plus siren, transformer, and 6 magnetic sensors w/wire. $249.

Zena micro engineering, Inc. 2205 26th Ave. South East Puyallup, WA 98373 (206) 848-4519

A CALL —A.P.P.L.E. UTILITY:

LCASE KILLER

By Eric Lambrecht

EVERYONE who has a lowercase-adapter wants to use it. But if you write software for other people (and some of them don't have a LCA), you are obliged to use upper case or to keep two versions of your program.

LCase KILLER solves this problem. When you write an Applesoft program, you can use lowercase as you please. And when the program has to run on an APPLE without LCA, just load it, BRUN LCASE KILLER and in an instant you have an uppercase version of the program.

LCase KILLER can also be of great help when you have a program with lowercase in it and you don't have a LCA. LOAD the program, BRUN LCASE KILLER and SAVE the "new" version under another name. So you will be able to work with the program and you still have the original version to be used when you buy a LCA later on.

The KILLER is very easy to understand. The Applesoft program is searched from beginning to end for lowercase characters (ASCII $61-$7A). If one is found, it is changed to uppercase by subtracting $20 from the ASCII value of the lower case character. And that's all !!! The program is complete relocatable, so it can be BRUN from any address you wish. The Applesoft program must be loaded in memory before BRUNning LCase KILLER.

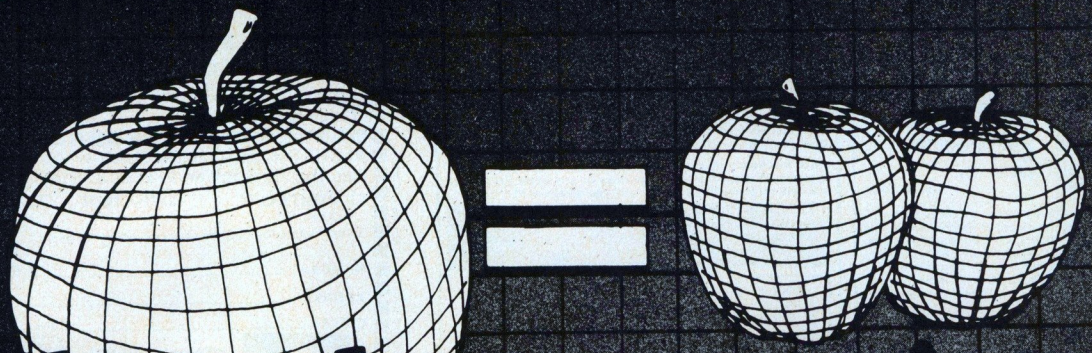
```
                    ********************************
                    *                              *
                    *          LCase KILLER        *
                    *                              *
                    *      by  Eric LAMBRECHT      *
                    *        Zulte, Belgium        *
                    *                              *
                    * Call -A.P.P.L.E. * Mar. 1982 *
                    *                              *
                    ********************************


                    ADDR      EQU   $06
                    PROGST    EQU   $67
                    BASIC     EQU   $3D0



                    *
                    *
                    *
                              ORG   $300
                    *
                    *
                    *
0300: A5 67                   LDA   PROGST      ;setup
0302: 85 06                   STA   ADDR
0304: A5 68                   LDA   PROGST+1
0306: 85 07                   STA   ADDR+1
                    *
                    *
0308: A0 01         ENDTST    LDY   #1
030A: B1 06                   LDA   (ADDR),Y    ;test for end
030C: D0 03                   BNE   SRCH        ;of program
030E: 4C D0 03                JMP   BASIC
                    *
                    *
0311: A0 04         SRCH      LDY   #4          ;skip line number
0313: B1 06                   LDA   (ADDR),Y
0315: F0 10                   BEQ   NEXTLN      ;end of line
0317: C9 7F                   CMP   #$7F
0319: B0 09                   BCS   NOLOW
031B: C9 61                   CMP   #'a'
031D: 90 05                   BCC   NOLOW
031F: 38                      SEC
0320: E9 20                   SBC   #$20        ;change to
0322: 91 06                   STA   (ADDR),Y    ;UPPER case
0324: C8            NOLOW     INY               ;continue
0325: D0 EC                   BNE   SRCH+2
                    *
0327: A0 00         NEXTLN    LDY   #0
0329: B1 06                   LDA   (ADDR),Y
032B: AA                      TAX
032C: C8                      INY
032D: B1 06                   LDA   (ADDR),Y
032F: 86 06                   STX   ADDR
0331: 85 07                   STA   ADDR+1
0333: 38                      SEC               ;for relocatable
0334: B0 D2                   BCS   ENDTST

--END ASSEMBLY--      ERRORS: 0      54 BYTES
```



hen does one Apple equal two Apples?

When you add J.O.Y.™

Sometimes one Apple isn't enough. Such as when you're sitting around waiting for your printer to finish its work before you can start your next program run.

But now you don't need a second Apple if you want to print and compute at the same time. Now you can use J.O.Y.

J.O.Y. is the software program that actually lets your Apple print and compute simultaneously. When you add J.O.Y. to your Apple, you'll be able to run programs from your keyboard at the same time your printer is working.

J.O.Y. is not a microbuffer. Its application is not limited to parallel printers. In fact, J.O.Y. works with any printer, card and slot. With DOS 3.3 and 48k of RAM, J.O.Y. can print text files or Apple-Soft program files. And it costs only a refreshing $39.95. (Pennsylvania residents add 6% sales tax.)

To order J.O.Y. for your computer, call us toll free at 800-233-7512. In Pennsylvania call 717-243-5373. Or write us at XPS, Inc., 323 York Road, Carlisle, PA 17013. We'll also be glad to send you our catalog of other innovative software, free.

Apple and Apple-Soft are registered trademarks of Apple Computer, Inc.

XPS, Inc.

Quality Microcomputer Software
323 York Road, Carlisle, PA 17013
(800) 233-7512 / (717) 243-5373

utility city

21 of our most-asked-for Apple Utilities
by Bert Kersey 48K Applesoft ROM

For YOUR Big Apple—Our 21 most-asked-for Apple Utilities on one big disk—**List Formatter** makes properly spaced & indented listings with printer page breaks. Each program statement is on a new line with If-Then's & Loops called out; a great de-bugger! **Catalog in any number of columns** & any page-width to CRT or printer. Automatically post the Run-Number & last-used Date in your programs. **Make any command invisible** in your listings; Access program lines in memory for garbage repair & "illegal" alteration; Quickly sort & store info on disk; Run any Applesoft file while another stays intact; Renumber to 65535; Save inverse, INVISIBLE & trick file names; Convert dec to hex & binary, or INT to FP; Append programs; dump the text screen to ANY printer . . . More too: **21 Programs Total!**

PLUS . . . APPLE TIP BOOK NUMBER THREE!
40 PAGES of new tips, tricks & articles—"DOS Trickery," "Copy Stoppers," "Programming the Reset Key" & much more. Plus understandable explanations of how each U-City Program works—One of the best Apple learning tools on the market today!

ALL 21 PROGRAMS **$29.50** +$1.50 Shipping

- ✓ **Utility City** on Applesoft Disk.
- ✓ Beagle Bros Apple Tip Book # 3
- ✓ Apple PEEKS, POKES & POINTERS Chart

alpha plot

Hi-Res Apple Graphics/Text Utility
by Bert Kersey & Jack Cassidy 48K REQUIRED

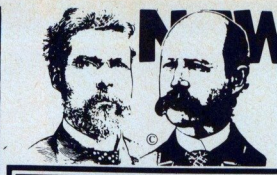
HI-RES DRAWING: Create hi-res pictures & charts, **appendable to your programs.** Keyboard or Paddle control; Optional Xdraw Cursor (see lines before you draw!); Any color mix or REVERSE (opposite of background); Circles, Boxes & Ellipses, filled or not. Bonus Programs too—**SCRUNCHER** stores hi-res in as little as 1/3 normal disk space. **SHIFTER** transfers any portion of the hi-res screen. Also superimpose hi-res images and convert Hi-Res to Lo-Res & back for fascinating abstracts!

HI-RES TEXT: Beautiful upper & lower case with Descenders; color or reverse; Positionable anywhere (NOT restricted by Htabs & Vtabs). Professional looking **PROPORTIONAL SPACING!** Adjustable Type Size, Leading (line spacing) & Kerning (letter spacing). Multi-directional typing; up, down, even backwards!

PLUS . . . APPLE TIP BOOK NUMBER FOUR!
40 PAGES of tips for making the most of your Apple's advanced graphics capabilities. Alpha instructions plus shape table tips, new titling & color tricks and fascinating animation experiments!

$39.50 +$1.50 Shipping

- ✓ **Alpha Plot** on Applesoft Disk.
- ✓ Beagle Bros Apple Tip Book # 4
- ✓ Apple PEEKS, POKES & POINTERS Chart.



NEW® Apple Utility Disks!

With Free PEEKS & POKES Wall Chart!

Beagle Bros MICRO SOFTWARE

4315 Sierra Vista
San Diego, CA 92103
714-296-6400

WITH EACH ORDER, YOU WILL RECEIVE . . .

- ☐ **Dos Boss, Utility City** or **Alpha Plot** on Applesoft Disk
- ☐ A 40-page **Apple Tip Book** (each entirely different)
- ☐ An 11x17 Apple **PEEKS, POKES & POINTERS** Chart

BONUS WITH EACH DISK!
POKE your Apple all night long with this handy 11 x 17 reference poster! The most useable PEEKS, POKES, POINTERS & CALLS scrounged up from every source imaginable! A great companion to our original Apple Command Chart.

dos boss

DISK COMMAND EDITOR
by Bert Kersey & Jack Cassidy

A classic utility you will ENJOY! **Rename Commands:** "Catalog" can be "Cat," etc. **Save-Protect your programs;** Unauthorized copy attempts produce "Not Copyable" message. 1-Key Program Selection from catalog (extremely handy!). Catalog Customizer—Change Disk Volume message to your title; Omit or alter file codes; Catalog by file type, etc. Rewrite Error Messages—"Syntax Error" can be "Oops!" or anything you want!

All of Dos Boss's change feaures may be appended to your programs, so that anyone using your disks on any Apple (booted or not) will be formatting DOS the way you designed it!

PLUS . . . APPLE TIP BOOK NUMBER TWO!
With valuable DOS information: "Poking Around in DOS," "Making Programs Unlistable," "Two-Sided Disk Tips," "Care of Your Drives," "Creative Peeking & Poking," "3.2 vs 3.3," etc.

ONLY $24. +$1.50 Shipping

- ✓ **Dos Boss** on Applesoft Disk.
- ✓ Beagle Bros Apple Tip Book #2
- ✓ Apple PEEKS, POKES & POINTERS Chart

MINIMUM 32K REQUIRED.

Ask around about Beagle Bros Software.
Our programs are practical, friendly & user-oriented, and always come with an amazing amount of documentation & juicy Apple information. If your dealer doesn't have our stuff, get on his case about it, or order directly from us. We ship all disks & books WITHIN 48 HOURS.

24 hour TOLL FREE Order Desk:
MasterCard, VISA or COD orders, call:
Nationwide: **800-854-2003** ext. 827
California: **800-522-1500** ext. 827
Alaska/Hawaii: **800-854-2622** ext. 827
(Orders only please. Questions, phone 714-296-6400. COD orders, add $3.)

Or order by mail (use this coupon or separate sheet)

(Use this coupon or separate sheet)

RUSH! The disks checked below
Plus The Tip Book and PEEKS & POKES Chart!

☐ **UTILITY CITY** ($31) ☐ **ALPHA PLOT** ($41) ☐ **DOS BOSS** ($25.50)

(Prices include First Class Shipping; Calif. residents add 6%.)

Specify: ☐ **3.2** ☐ **3.3**

NAME$= _____

ADDR$= _____

CITY$= _____ **ZIP=** _____

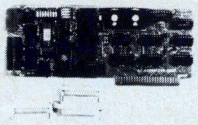
VISA **VISA/MASTERCARD: Include Account No. & Expiration Date.** master charge THE INTERBANK CARD

Mail U.S. check, money order or Visa/MC Numbers to: **BEAGLE BROS, Dept. Z**
4315 Sierra Vista,
San Diego, CA 92103
Telephone (714) 296-6400.

Please add $4.50 for shipping outside US or Canada.
COD orders, add $3. California residents, add 6%.

NEW * APPLE II PRODUCTS

* DUAL COMM PLUS™
TWO SERIAL PORTS

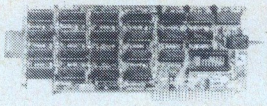


NEW

- Two fully independent serial I/O ports
- Combines the features of the APPLE HIGH SPEED SERIAL CARD™ and the APPLE COMMUNICATIONS CARD™ in one card (with handshaking)
- Thumbwheel switches select the Apple slot locations. Sixteen switch selectable baud rates for each port (50 to 19200 baud)
- On-board firmware provides extensive printer and U/L case terminal/modem support. Great with an 80 x 24 card and a Novation CAT or Hayes Smartmodem
- Can be programmed for async., sync., or even SDLC operation. Supports interrupts. Uses Z80 SIO chip.

LIST PRICE: $239.00
MANUAL ONLY: $15.00

* MEMORY PLUS™
16K RAM EXPANSION

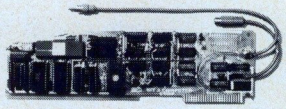


NEW

- Add another 16K to your 48K APPLE
- Works with PASCAL, DOS, CP/M, FORTRAN, COBOL, PILOT, Personal Software's VisiCalc, INTEGER and APPLESOFT BASIC. Many other Apple software packages benefit from the additional storage provided by the MEMORY PLUS card.
- Three LED's show memory select and read/write protect status
- Toggle switch to shut the card off or to select between the standard monitor ROM or the on-board firmware socket. The firmware socket can hold an alternate Apple monitor ROM or a 2716 program (user provided).

LIST PRICE: $149.00
MANUAL ONLY: $15.00

FULL-VIEW 80™
80 X 24



- Permits selection of 80 column or Apple 40 column/graphics on a single monitor via keyboard or program control
- 8 x 10 cell size upper/lower case character set with full lower case decenders
- User definable EROM character sets via 2716 (127 chars) or 2732 (255 chars)
- On-board 2K firmware supports HTAB, VTAB, U/L case keyboard, DATAMEDIA CONTROL CODES, INVERSE/NORMAL VIDEO, and a special printer driver
- Low power requirement
- Easy installation - just plug it in!
- Compatible with many 80 column word processors
- Compatible with the D.C. Hayes MICROMODEM™

LIST PRICE: $379.00
(2716 std. char set)
MANUAL ONLY: $15.00

All Bit 3 Apple products features:
• Easy plug-in installation • Operation with PASCAL Z80 SOFTCARD™, APPLE II and APPLE II PLUS
• 50/60 hz operation

VISA® MasterCard®

BIT 3 COMPUTER CORPORATION
8120 Penn Avenue South • Suite 548
Mpls., MN 55431 • (612) 881-6955

Apple II is a registered trademark of Apple Computer, Inc.; Z-80 Softcard is a registered trademark of Microsoft Consumer Products.
D.C. Hayes Micromodem is a trademark of D.C. Hayes, Associates, Inc.; Novation CAT is a registered trademark of Novation, Inc.

A.P.P.L.E. MEETING MINUTES

By David A. Lingwood/Secretary

December 15, 1981
Computer Empire, Burien

PRESIDENT Dick Hubert called the meeting to order at 7:45PM, with 53 members and guests in attendance.

Old Business: None.

New Business:

Dick announced our first major hardware product for club members: the AP.Drive Apple-compatible disk drive (sans controller) at an introductory price of $379. It will be available February first. Price will become $395 after March first. The unit comes with a 120 day warranty, with direct shipment to the manufacturer for repairs. An optional technical manual is also available.

Announcements:

Warren Raymond, of Custom Industries, a Bellevue sheltered workshop, needs volunteer programming help with a piece-rate payroll program. Custom Industries also does assembly and packaging for hardware and software.

Committee Reports:

Bob Huelsdonk, chair of the Nominations Committee, reported that all incumbent officers have agreed to be re-nominated. Mike Deville has also been nominated for one of the Director slots. Nominations were opened to the floor, and then closed for lack of any further nominations. It was moved and approved to accept the current names in nomination.

Dick Hubert gave the Treasurer's report, noting a $10K positive cash flow last month. The newly increased membership and advertising rates will cover magazine production costs, and bring *Call* -A.P.P.L.E. in line with similar publications. We are are also approaching our 10,000th member number, and will come up with some sort of award.

A member asked about ground rules for writing for *Call* -A.P.P.L.E. Author guidelines are available from the office, Val Golding, or the Secretary.

The business meeting was adjourned at 8:20PM.

Q&A

Mike Thyng began with a review of the as-is software volume #45, containing Applesoft business software. The income tax program is thin but helpful, and the payroll program may be useful as a base from which to modify. As is #144 contains the Eamon adventure games, proliferating onto disks 144 and 164.

On to questions. A query from Jim Janney (824-4013) on braille or other output media for the blind and/or deaf produced a lead to the Pacific Science Center, which held a festival some six months ago dealing with the handicapped. Contact Jim if you have more information or a similar interest. After some discussion of computer applications for the handicapped, a proto special interest group was formed. More on this as the group gets in gear.

Other questions covered loading shapes from tape, floating point double precision, and the problem some have had with green screen monitors affecting disk drives placed immediately beneath them (move the monitor!).

Speaker

Do your disks go brrrrrp! in the night? Guests from Verbatim Corporation discussed the maturation, design, production, care and feeding of diskettes. The problems with "back-siding" disks were discussed (the pad on the normally unused top side loads with oxides and other crud, and may abrade the prime recording surface if the disk is flipped; also, the top side may or may not record perfectly). Radial scratches and temperature extremes are also to be avoided. Piles of jacketless disks invite problems from dirt, dust, and jacket warpage. Remember to copy disks created on mis-aligned drives before replacing the unit or having it repaired. Airport magnetometers have not been known to produce problems with disks. The future seems aimed at even greater reliability and higher recording densities.

January 19, 1981
Computer Empire, Burien

The meeting was called to order by President Dick Hubert at 7:35PM. There were 110 members and guests present.

Old Business:

Minutes of the October and November meetings were approved as published in *Call* -A.P.P.L.E.

New Business:

The Treasurer's report shows $102K in receipts, 73K expenses, and $100K banked. Our yearly cash flow is approximately one million dollars.

Announcements:

Dick Hubert announced "What's Where in Apple," and "The Power of VisiCalc," new book products available through the club to members.

Mike Lane (385-4081) is looking for programs or sources on health, diet or nutrition.

The business meeting was adjourned at 7:50PM.

Q&A

This month's installment of Mike Thyng's review of the as-is software covered #100, a games disk containing several decent games.

The questions covered the usual gamut: Why a 3.3 file longer than 122 sectors would clobber the T/S list (no good answer); sources of time sharing service (e.g., Source, CompuServ, etc.); and when ProFile will be available for the][(probably by April or so, with the rumored "3.4" DOS).

Speaker

Here we go (went) again — through the information-manic future with rod, gun, and cryogenic outer space computer — courtesy of tour guide Jonathon Post. Not that the future is any more formed than when Mr. Post first entertained us in Lynnwood last year, but his eclectic style and presentation may well help shape it. (I swear the man could speak from a shopping list — "A Canticle for Post?" — hmmm.) If anyone asks, just say your work with the Apple is a survival training school for the information = power future.

Supercharge Your APPLE II*

The Axlon RAMDISK™ 320K Memory System for the Apple II and Apple II Plus* provides access speeds never before available. The Axlon memory system is designed to interact with Apple DOS 3.3* and Apple Pascal 1.1* like two standard floppy disk drives while delivering the lightning fast access speeds of RAM memory. This also leaves 32K of RAM for advanced programming techniques. The interface board is slot independent and draws no power from your Apple. The rechargeable battery system built into the unit provides three hours of backup in the event of a power loss. Drop by your local Apple dealer or contact Axlon, Inc. for more information.

* Trademark of Apple Computer, Inc.
* Pascal is a Trademark of U.C.S.D. Regents

- Plug-in compatibility
- 320K bytes of RAM (200NS) memory designed to function like two 35 track floppy disk drives
- Compatible with Apple DOS 3.3 and Apple Pascal 1.1
- Same size as the Apple Disk II* Drive
- Invisible memory refresh - even with the Apple turned off
- Rechargeable battery system built-in to provide 3 hours of auxiliary power
- Slot independent interface board - draws no power from your Apple
- All firmware is in static RAM on the interface board
- Includes software for diagnostic, fast load and copy routines, and business applications

AXLON

170 N. Wolfe Road,
Sunnyvale, CA 94086
(408) 730-0216

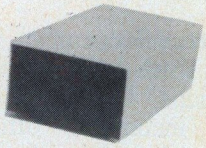
apple

DISK FOR APPLE II OR APPLE II PLUS

KING COBRA

NOW YOU CAN HAVE ANOTHER DISK DRIVE FOR UNDER $320.00

KING COBRA Disk Adapter Kit
-converts the Shugart SA400 * to full compatibility use with the Apple!!
-100% operation compatible with all Apple functions, software, etc!!!
-takes less than 1/2 hour (if you use your own SA400)
-takes less than 10 minutes with the jumper SA400 below!!!
JUST PLUG IN A SMALL ADAPTER BOARD!!!
-the kit includes:
-Adapter board and assembly instructions (VERY EASY!)
-ALL Cables!
-CUSTOM COLOR-COORDINATED DISK DRIVE CABINET (Looks like the Apple)
-THE KIT IS FULLY GUARANTEED FOR 60 DAYS!!



THE EXACT SAME DRIVE MECHANISM AS THE APPLE DISK DRIVE!!

ONLY $79.95!

Due to increased power requirements only one SA400 should be used per system.

Systems already stressing the power supply may have trouble using this drive (PLEASE CALL IF YOU HAVE QUESTIONS).

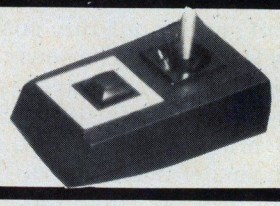
Jumpered SA400- Only $240.00!
Regular SA400- Only $230.00!
*Shugart SA400L disk drive

COBRASTICK Dual Joysticks >>ONLY $34.95<<

KING COBRA

-handsome jet black cases with brushed aluminum sticks
-compatable with any software and I/O extension unit
-longlife switches
-Our DUAL costs less than most SINGLE joysticks!!!
-a flexible, powerful, control system for computer input for exciting, competitive games!

SINGLE COBRASTICK Joystick, with one button input, >>$19.95



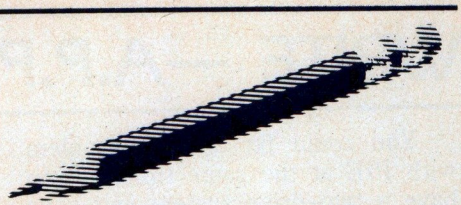
TO ORDER: Send check or money order or call (312) 393-4385

RJ ELECTRONICS LABORATORY
Box 186 Naperville, IL 60566

Add $3.00 for handling and insurance
Add $6.00 for handling for SA400's or Disk Adapter
25% deposit required on COD orders over $100
Please allow 4-6 weeks before delivery

Apple and Apple II are trademarks of Apple Computer, Inc. • DEALER INQUIRES INVITED • IL residents add 5 1/4 % sales tax

WRITE -A.P.P.L.E.



By the Readers

Dear Val,

As someone interested in assembly-language programming, I was very glad to see the article "Three Macro-Assemblers" by Jonathan Levine in the November/December issue of *Call -A.P.P.L.E.* We assembly-language programmers are definitely in the minority, and it's always refreshing to see something aimed at us — and especially reviews of products that help us develop more software.

I'd like to add one more assembler capable of using "macros" to the three covered in the article. It's the S-C Assembler Version 4.0. I've been using this piece of software extensively for the past year; I've found it to be extremely reliable, and I have yet to use all of its capabilities.

The S-C assembler allows the inclusion of macros through its ".IN" directive. Like Big Mac, it doesn't permit nested macros. It supports assembling to memory, assembling to a named disk file, or assembling code "org'd" at one address but stored at another address of my choice. Most of the DOS commands are directly available from the keyboard, so the user can easily run catalogs, initialize disks, and use files in much the same way one does when developing BASIC program.

As Jonathan Levine stated in his article, support is a definite consideration when purchasing an assembler, and the S-C assembler has considerable suppsort. At this time there are two intelligent dis-assemblers, a user-definable formatting utility, a global editor utility, and a symbol cross-referencing utility available at a reasonable cost. In fact, you can even buy the source code for the assembler itself (on disk, ready to assemble) for less than a hundred bucks!

Bob Sander-Cedarlof is supporting his assembler (and all Apple assembly-language programmers) with a unique monthly publication he calls the "Apple Assembly Line." Within its pages Bob includes hints, tricks, and some very valuable information on how to use the internals of the Apple. During the past year, articles (with accompanying source listings) have appeared on using Applesoft floating point and Hi-Res routines from machine language, disassembled portions of DOS, and routines that enhance the S-C assembler itself, giving it a block move and copy function and a PLE-like line editor. Every three months, Bob releases a disk with all the source code from the past three issues, compatible with his assembler. The annual subscription rate for the "Apple Assembly Line" is $12.00; each source disk is $15.00.

In summary, the S-C assembler is well worth its $55.00 price tag, and I think it warrants being added to the list of assemblers appearing in the article. S-C Software's address is: PO Box 280300, Dallas, Texas 75228.

Don Taylor
PO Box 1736
Poulsbo, WA 98370

Dear Val,

Your columns by Cornelis Bongers and Charles Rusch are very enjoyable. The $48 Trap has always puzzled me and it was a treat to read Mr. Bongers' careful and thorough analysis.

Mr. Rusch may be interested in a problem with PIE. Users who exit PIE via Format (instead of exiting with the Quit command in the PIE command processor) will find they have to re-boot. You can use the following information any way you like.

The reason is that PIE changes the DOS ASCII command table for the Load and Save instructions so that DOS doesn't recognize the commands "LOAD" and "SAVE" anymore. After exiting, DOS no longer recognizes Load and Save and passes them on to Applesoft. Applesoft sees these as cassette commands and the Apple hangs.

DOS intercepts all commands to the PIE command processor just like it does when Applesoft is running. PIE does this so commands such as Delete and Unlock will function without PIE intervention. Since PIE has to convert Load and Save to Bload and Bsave, (PIE files are binary), PIE fixes DOS to reject Load and Save as non-DOS commands.

In fact, PIE changes Load to Qoad and Save to Qave but restores them when exiting the command processor with the "Q" command. However, PIE fails to restore the Load and Save commands when going to Format. You need to patch PIE to restore and fortunately there is unused memory at the point where you must patch.

The command to Brun Format starts at $125E in PIE. To correct the problem, go into the Monitor and type:

```
*125E:A9 4C 8D 88 A8 A9 53 8D 8C
A8 20 A1 08 8D 84 C2 D2 D5 CE C6
CF  D2  CD  C1  D4  8D  00
```

Here is a disassembly of the patch: the patch:

```
LDA  #$4C"L"
STA  $A888Restore Load com-
       mand in table
LDA #$53"S"
STA  $A88CRestore Save
       command in table
JSR  $08A1Print routine
       (prints following
       string up to $00)
HEX 8D 84Prints carriage
       return, control-D
"BRUNFORMAT"Print DOS
  command
HEX 8D 00End with carriage
       return and signal end
       of print string with a
       "00"
```

WRITE —A.P.P.L.E.

You could fix another problem with PIE at the same time. If you have a two-drive system and have been working on a file from the second drive, then PIE will try to load Format from the wrong drive. There is room in PIE to add ",D1" after "BRUNFORMAT" so PIE will always go back to drive 1 to load Format.

I did a lot of other things to PIE, like modifying it to work easily with my Corvus hard disk, adding commands to Format to build and print out an index and other modifications to make an excellent package even better. I won't give these away, though, but would be willing to consider a fair price.

Thanks again for publishing two fine columnists.

Peter G. Bartlett, Jr.
529 West Belden Avenue
Chicago, IL 60614

VANDAL FOILERS FOILED, EDITOR OMITS PROGRAM

```
10  REM

 Applesoft Vandal Foiler
   by  Leighton Paul

Call -A.P.P.L.E. Feb 1982

100  TEXT : HOME
200  POKE 42061,76: POKE 42062,14
     1: POKE 42063,183: POKE 4698
     9,133: POKE 46990,214: POKE
     46991,108: POKE 46992,96: POKE
     46993,157: POKE 43140,80: POKE
     43141,82: POKE 43142,79: POKE
     43162,127: POKE 43214,127
300  PRINT "VANDAL FOILER is now
     installed in the": PRINT "DO
     S memory area.": PRINT
310  PRINT "INIT a blank disk wit
     h 'PROT HELLO',"
320  PRINT "then copy programs as
      desired to the"
330  PRINT "protected disk.": PRINT

340  PRINT "VANDAL FOILER will be
      activated with the";
350  PRINT "first 'LOAD' or 'RUN'
      command."
400  REM

Modifies 48k 3.3 DOS
```

WRITING THE WRONGS

By the Wrongdoers

A set of parenthesis was omitted from *Our Logical Apples* on page 19 of the February *Call —A.P.P.L.E.* Line 30 should have read:

```
Z = (A>10) * (A>20)
```

For A.P.P.L.E. Members Only

An A.P.P.L.E. Breakthrough

Ap. Drive

The First 100% plug Compatible Disk Drive for The Apple II* or Apple II+*

Plugs Into Any Apple Disk Controller Card

Member Price:

$395.00

Complete Technical Manual with Schematics: $30. Additional

Overseas shipping wt. 6 lbs.

Both for $475.00

Ap. Controller

for the Apple II* or Apple II+* boots on either 13 or 16 sector DOS

Apple DOS must be supplied by user

Member Price:

$95.00

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

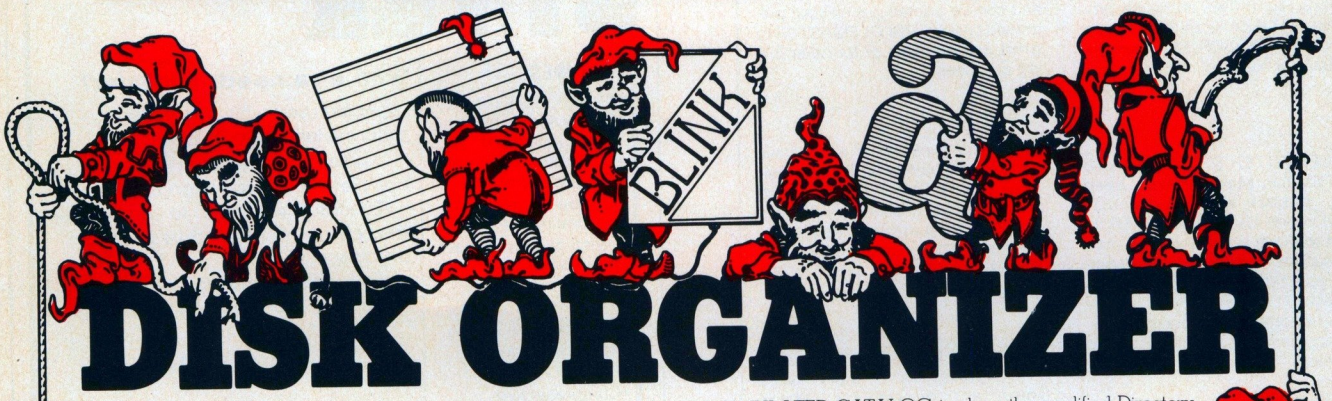
*Apple II and Apple II+ are registered trademarks of Apple Computer, Inc.

Washington state residents add 6.6% sales tax.

All foreign orders (incl. Canada) will require additional postage and shipping.

Organization a problem?

Look what Sensible Software is doing for you...



DISK ORGANIZER

Here is a fast and friendly utility to organize the files on your diskettes using DISK ORGANIZER. You can order the files on a diskette not only alphabetically, but in any order you want, TITLES can be entered, describing a file or group of files on the diskette. These TITLES may be entered in normal, inverse, flashing, and lower case letters. All work is done on a RAM version of the Directory; so if you decide the changes you made are not correct, you can simply abort the process and keep the old catalog. Since DISK ORGANIZER works on a RAM version of the Directory, disk access is kept to a minimum; hence, DISK ORGANIZER is extremely fast.

The following functions are also included:

- DYNAMIC DISPLAY of all file names in the Directory.
- RENAMING the "HELLO" program.
- A SIMULATED CATALOG to show the modified Directory before it is written to the Diskette.
- ALPHABETIZING the file names.
- UNDELETING deleted files.
- PURGING deleted files.
- RENAMING files (with the same character input options as TITLING.
- LOCKING and UNLOCKING (some or all) files.
- DELETING files.
- DELETING DOS for increased data storage.
- A powerful SMART KEY to automatically locate the next valid file name for any specified operation.

DISK ORGANIZER...Apple II or Apple II Plus with 48K and one or more Disk Drives...**$30.00**

Is inter-disk travel and intra-disk file manipulation causing difficulties?

Then SUPER DISK COPY is for YOU!

Super Disk Copy is much more than just another copy program. It's practically a 'mini-DOS' in itself. Super Disk Copy is completely menu driven and works with DOS 3.1, 3.2, and 3.3. Files may even be transferred from one DOS to another (example: DOS 3.3 to DOS 3.2). Super Disk Copy makes the conversion to DOS 3.3 less painful. After seeing Super Disk Copy, you will agree it is the best copy program on the market! Among the many features are:

- Wildcards may be entered anytime a file is used. They allow both automatic or user verification of the file handling. (Example: to copy file names that start with 'Apple', you would enter 'Apple = ').
- COPY single files (Applesoft, Integer, Text, Binary, Relocatable).
- COPY DOS.
- COPY ENTIRE DISK. Super Disk Copy uses a unique technique for fast copying.
- A BRUTE FORCE COPY made for Pascal, FORTRAN, and CP/M diskettes.
- An optional rearrangement of files so that they occupy contiguous sectors for improved access time.
- UNDELETE deleted files.
- A PLOT of disk usage.
- FIX file sizes. Make sure your valuable files are protected and unused sectors are available.
- DELETE DOS. Frees up extra space on your diskettes.
- ALPHABETIZE file names in directory.
- REPLACE illegal characters in file names.
- Optional 'INIT' of copy diskette.
- LOCK or UNLOCK files and much, MUCH more...

SUPER DISK COPY was one of only two programs to receive a rating of 100 in PRICE/USEFULNESS RATIO in THE BOOK OF APPLE COMPUTER SOFTWARE 1982. They said "This is a definite MUST HAVE utility package."

48K and DISK II required...**$30.00**

For a complete catalog send $1.00, refundable with your first purchase.

If you have a large software collection and have problems locating specific programs or you need to be reminded of what they do...then you need MULTI-DISK CATALOG

Multi-Disk Catalog is designed specifically for keeping track of the contents of the APPLE diskette library. The resulting master catalog can be sorted, searched, and printed. Multi-Disk Catalog is entirely menu driven, easy to use, and very fast. Some of the unique features included are:

- Works with 13 and 16-sector formated diskettes (DOS 3.1, 3.2, and 3.3). Diskettes from either type DOS can be added to the same data-base used.
- You assign the volume identification number (range 1 to 999); there is no need to INIT each of your disks with a unique volume number.
- A 2 letter classification code can be added to each file name. The classifications are retained when the disk volume is updated.
- Multi-key sorting. Up to 3 keys may be selected for sorting the file names. The sort keys include disk number, file name, classification, file type, and file size.
- Multi-Disk Catalog is written entirely in machine language, so it is very fast.
- A powerful search mask can be used to find a file name or groups of file names.
- Multi-Disk Catalog supports special printer interfaces.
- Both sides of a diskette may be entered under the same disk number.
- Titles may be entered for each diskette and much more!

"There is no doubt about it, this is the best and most versatile master catalog program available"—THE BOOK OF APPLE COMPUTER SOFTWARE 1982.

48K and one Disk Drive required...**$25.00**

Sensible Software

6619 Perham Drive Dept. C • West Bloomfield, Michigan 48033 • (313) 399-8877
Visa and MasterCard welcome, please add $1.25 postage and handling per diskette.

TWO FOR ONE!

Buy FILEMASTER and receive two complete programs.

That's Right! Two complete programs - DATA-BASE SYSTEM and a WORD PROCESSOR. Use them individually or together.

FILEMASTER'S DATA-BASE SYSTEM can **store records, keep lists, project sales** and much, much more. Use FILEMASTER to **print mailing labels** for large mailings such as Christmas cards. **Inventory** your tape or book libraries on FILEMASTER. **Index** your magazines. Important telephone numbers and appointments can be produced with ease.

Set up a **checkbook register** and let FILEMASTER caculate the balances. Simple **payroll** can be done and can be printed out on your preprinted checks. Use FILEMASTER to **print out invoices** or for other billing purposes.

Are you a collector? Keep close track of your **collectables** and their value with FILEMASTER. Also keep and inventory household valuables for insurance purposes. The applications of our program are virtually unlimited.

Use FILEMASTER'S WORD PROCESSOR as a **diary.** Store important documents. **Write letters.** Store business notes. Use it in conjunction with FILEMASTER'S DATA-BASE SYSTEM and you can send out **form letters** to each name on your mailing list.

FEATURES:

- Machine Language
- Instantaneous Sorting
  over 1000 pieces of data per second, forwards or backwards, alpha-numerically or by number value.
- Instant Search
  single character, single word, group of words, whole or parts of files.
- Instant Access to all Records
- Easy Update or Correction of Files
  add, delete, rename or reorganize fields at any time; move an entire block of records.
- Divide and Merge
  divide a large file into two smaller ones; merge two small files into one large one.
- Simple Mathematical Functions between Fields
  payroll, checkbook, sales projections, ect.
- Versatile Report Generator
  format your own report; fill in pre-printed forms, invoices or checks.
- Word Processor
  capitalization, indentation, insert and delete characters, right and left margin justification, variable line spacing.
- Holds 9 Files per Diskette
- Up to 20 Fields per File
- Works with most Printers
- One Disk Drive
- 48K Applesoft in ROM
- DOS 3.2 or DOS 3.3
  (please specify with order)

NOW THRU MARCH 31st

Sale $119.00*

Postmark your order or call by March 31, 1982 and receive FILEMASTER for **$119.00.*** You will promptly receive a copy of the FILEMASTER diskette and documentation in a hardbound binder for safe storage.

After the sale, FILEMASTER will return to its regular retail price of $149.00.* SO ORDER NOW.

To order Visa or Mastercard call (517) 337-7423

Michigan Residents Add 4% Sales Tax

Businessmaster Inc.

6443 COLEMAN RD. • E. LANSING, MI 48823

BUG BYTER

Bill Budge, creator of "Raster Blaster," needs it:

"...I'll never write another program without BugByter!"

FEATURES:

- All registers displayed
- Compatible with all Apple languages
- Completely relocatable
- Full hex and ASCII I/O
- Multiple options while in trace mode
- Literal and transparent breakpoints
- Resident assembler
- Resident disassembler
- User-definable screen
- Ram screen dump in hex and ASCII
- Comprehensive documentation
- Single keystroke operation
- Instructional cycle counter
- Hexadecimal/decimal conversions
- Can run in add-on ram card

Who else needs BugByter?

...Apple* users who want to learn machine language.

...Apple programmers in need of a complete 6502 debugging tool.

...Educators who need to demonstrate the operation of the Apple's central processor.

...Software professionals who need to display and control all 6502 registers.

BugByter
is
NOW AVAILABLE AT $39.95
on diskette for Apple II or Apple II+
from
COMPUTER-ADVANCED IDEAS, INC.
1442A Walnut Street, Suite 341
Berkeley, CA 94709
(415) 526-9100

*Apple is a registered trademark of Apple Computer, Inc.

CLASSIFIED

(APPROVED FOR RELEASE TO APPLE USERS)

"You put *what* in your Apple computer?"

Investments. Account data. Wills. Phone lists. Sooner or later you will create a file into which you will write some very confidential information. And that's why Passage Research has developed a special utility that will encrypt that DOS 3.3 file through software routines rather than the expensive (and conspicuous) black-box approach. Now you have the means to keep your private affairs private. Totally private.

The bits and bytes of your file will get hopelessly mixed up according to a cipher "key" that you keep as secret as you want. Over one hundred million billion different keys are available for selection. (This many keys may sound like a lot, but it's typical of a modern encryption process.)

To make your encrypted file understandable again, all you have to do is reactivate the appropriate key schedule and run the file back through. It's easy.

With this utility (and a few easy-to-learn mnemonic commands) you can create a personalized encryption algorithm that is specially tailored to your needs. If you want, you can execute successive encryptions with different keys and then strip off the outer layers, one by one, to reveal the original text. And you can call many routines from your own application programs to do "codebook" encryption (documentation included).

Passage Research

945 Turquoise St., Suite G
San Diego, California 92109

$39.50 postpaid. California residents add 6% sales tax. Specify Software Package No. U-17 for Apple II or Apple II+. Memory requirement: standard 48K. Due to time required for checks to clear financial institutions, please allow 3-5 weeks for disk delivery. Software documentation shipped immediately upon receipt of order.

(Apple is a registered trademark of Apple Computer, Inc.)

A.P.P.L.E.
ANNOUNCES:
Big Mac.LC Macro-Assembler/TED†
for the language card or RAM card

The most advanced editor-Assembler available for the Apple II* or Apple II+*

Runs on the Slot 0 RAM card and supports all of the following:

- Full nestable Macro capabilities with local variables and conditionals
- Correct assembly of all Sweet 16 opcodes without register equates
- Conditional assembly with DO, ELSE, FIN
- PUT Pseudo-Op allows file chaining
- 34 psuedo-ops
- Flash and inverse strings allowed
- List with or without line numbers
- Optional delete when copying lines
- Edit string literals directly
- 11 edit mode commands modeled after Program Line Editor
- Data can be specified in hex, decimal or binary
- Labels support four arithmetic and three logical operators
- Reads, edits and writes text or binaryfiles
- Companion disassembler handles Sweet 16, assigns monitor labels
- 80 column card compatible
- 64 page instruction manual
- And more!

Fully user oriented, "Big Mac.LC" Macro Assembler/TED contains all of the sophisticated features you would expect in commercial products selling for $50.00 and up, yet is designed for easy use by the beginner.

INTRODUCTORY PRICE:

$27.50

Available to A.P.P.L.E. Members only!
Standard version only $22.50

†Available about April.

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055

Overseas shipping weight: 4 oz

*Apple II and Apple II+ are registered trademarks of Apple Computer, Inc.

All foreign orders (including Canada) will require shipping and handling charges

A.P.P.L.E. ORDERS
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514

I am enclosing my check/MO for $ _____

My membership number is: _____ exp. date _____

Please send me the following: _____

_____

_____

_____

Name: _____

Address: _____

City: _____ State: _____ Zip Code: _____

Phone: _____

MasterCard acc. # _____

VISA® acc. # _____ exp. date _____

Washington state residents must add 6.6% state sales tax.
All foreign orders will require shipping and handling charges (including Canada).

AD INDEX

A.P.P.L.E..... 14, 16, 31, 41, 44, 57, 68, 76, 79, 80
Advanced Logic Systems 8
Advanced Operating Systems 67
Amdek ..... IBC
Andromeda Inc. 64
Axlon, Inc. 74
Beagle Brothers 72
BIT 3 Computer Corp. 72
Brøderbund Software 21
Businessmaster, Inc. 78
Bytes & Pieces 54
Cases, Inc. 7
CET Research Group, Ltd. 10
Computer-Advanced Ideas 78
Computer Consulting Services 35, 38
Connecticut Information Systems 20
Datamost 5, 56
Dickens Data Systems 6
Discount Data 29
D. R. Jarvis Computing 19
Futra Company 12
Krell Software Corp. 25
Leading Edge Products, Inc. ..... IFC
Madwest Software 17
Micro Lab 46, 47
Micromate Electronics, Inc. 39
Orange Micro 1
Passage Research 78
Professional Computer Systems 11
Prometheus Products, Inc. 4
R. J. Electronics Laboratory 74
Sensible Software 30, 40, 77
SGC 52
S & H Software 39
Software Solutions 19
Southeastern Software 26
Southwestern Data Systems 60, 61
Spectrum Software 36
Strategic Simulations, Inc. 32
Strategic Software 35
Street Electronics Corp. 69
Sympathetic Software 58
Synergistic Software ..... BC
Tess Enterprises, Inc. 51
Westware, Inc. 14
XPS, Inc. 50, 71
Zena micro engineering, Inc. 69

Now, for the first time, a book on 6502 Assembly Language programming for the Apple, specifically directed to the beginner.

Programming 6502 Assembly Language
by Randy Hyde

Member Price

$17.00

Overseas shipping weight: 12 oz.

A.P.P.L.E. Orders
304 Main Ave. S. Suite 300
Renton, WA 98055
(206) 271-4514
Visa, M/C welcome

For A.P.P.L.E. Members Only

Washington state residents add 6.6% sales tax.

All foreign orders (incl. Canada) will require additional postage and shipping.

UNCLASSIFIEDS — A.P.P.L.E. ORDERS

CALL -A.P.P.L.E. (un)Classifieds

Members **Non-commerical** rate .05/word
Members commerical rate .25/word
All others: 1.00/word, $25 minimum
Send ad copy with payment to editorial office

Put an ATLAS in your APPLE! Now you can quickly locate North American cities and determine exact distances between them. Includes a 411 city Data Base and Hi-Res MAP. Ten day moneyback GUARANTEE! ONLY $24.00. DOS 3.2/3.3. For FREE information or order write: Micro-ATLAS. Columbia Software, Box 2235, Columbia, Maryland 21045. (301) 997-3100.

WANTED: Humorous lines for computer-related comic strip by artist with solid syndicate contract. Make a hobby of collecting computer jokes, and get paid $5 for each used. Write for format to L. M. Boyd, Crown Syndicate, Inc., 5 Crown Road, Weatherford, TX 76086.

NEW: LOWER CASE keyboard conversion kit. Never anything like available before! Convert your Apple II or II+ keyboard into a TRUE typewriter like keyboard. Input upper and LOWER CASE characters. TOUCH-TYPE just as you would on a typewriter. Essential for WORD PROCESSING!! No pre-ESCape needed for upper case as in current method in wide use today. Ability to input all 128 ASCII character codes. Apple ESCape code cursor functions still available. Minimal hardware installation is fast and easy. Some Apples may require removal of case (screwdriver needed). Software interface routines on diskette for Applesoft, Integer, and Apple DOS. Several options included. $10.00 includes postage, handling, and any tax where applicable. Specify DOS 3.2 or 3.3. Cassette tape users request program listings instead of diskette (same price). Send $10.00 check or money order to: *** LOWER CASE, POB 1978, Oroville, CA 95965 *** Please allow 6 to 8 weeks for delivery. Send SASE (!!) for more information.

48K APPLE II, Disk Drive, Silentype, Software, Call (206) 922-0437.

ADVENTURE SERIES Jack and the Beanstalk Adv. is offered free to introduce the series. Send a disk and $2 for postage or send $6 (and no Disk). DISK 1 contains The Three Pigs and The Wolf Adv. & The Thief of Baghdad Adv., while DISK 2 contains Robin Hood Adv. & Red Ridding Hood Adv. Disk 1 and 2 (48K APPLESOFT) sell for $24.95 each. Each of these programs places you as a character in one of these classic stories. CONTACT: M. A. C. SOFTWARE, P.O.B. 27 Chillicothe OH 45601.

ATTENTION HAMS! Tired or preparing QSL's by hand? I was so I created customized File Cabinet. Brand new program maintains ham log PLUS prints QSL's automatically using information from log. Quick, effortless and accurate. Impress DX with your unique QSL's! Designed specifically for Apple II Plus with disk and any printer. First quality disk plus complete documentation. Send $35.00 to Hank Lefcort,, N1BMZ, 131 Round Lake, Ridgefield, CT. 06877.

WANTED — Carbon Dioxide sensor (0-2% in air) to input to A/D board. If you know of supplier, new or surplus, PLEASE write: Bruhns. Box 395. Pinebush, NY 12566 (914-361-2833).

SIGHT 'N SOUND — from stereo or cassette player is synchronized with light show of user's design on Hi-Res screen. Demos include kaleidoscope, EKG, and guitar player. 20 page manual. 48K & Applesoft req'd. No hardware modifications — $26.95. Compugraphics Software, #6 White Plains Dr., Chesterfield, MO. 63017 or call (314) 532-5377.

HELP YOURSELF — with the new OMNIWARE line of inexpensive high quality applications software for the APPLE II (Applesoft, 48K RAM, DOS 3.3). Programs now available include OMNIFILE, a full featured file manager and report generator ($49.95); OMNITREND, a powerful multiple regression trend analysis program with statistical calculations and extensive hi-res graphics ($44.95); and OMNIGRAPH, a flexible data plotting program that allows X-Y plots, bar charts and pie charts ($39.95). All three are available as OMNIPACK for $99.95. Also available is the OMNITEST educational system, which allows you to build your own question and answer files on any subject, then review them in drill or quiz game mode ($29.95). Introductory prices are good until 5/1/82, and all programs come on unlocked diskettes with listable code. Educational Computing Systems, Inc., 106 Fairbanks Road, Oak Ridge, TN 37830. (615) 483-4915.

PEACHY WRITER — New! A simple, easy to use text editor at a price you can afford! $24.95. HAPPY FACE — 4 hi-res games for children: Riddles, Famous Sayings, Nursery Rhymes, Spelling. 48K $15.00 COLLEGE PHYSICS — 11 disks, 75 programs, extensive graphics $200. AQUARIUM 25; DINOSAURS $25; CHRISTIAN EDUCATION $15. Write for a free catalog. VISA M/C. Cross Educational Software, Box 1536, Ruston, LA 71270 (318) 255-8921.

APPLE PARALLEL PRINTER CARD — Sold my printer but not the interface card! Like new; used 6 months. $95 or best offer. Rick Hughes, 115 Country Club Drive, Florida, NY 10921. (914) 651-4938.

For A.P.P.L.E Members Only

DISKPAK 12 - $16.50

Newly Revised for Apple II+*

A collection of 40 Applesoft games, 8 color demos, etc.

A.P.P.L.E. Orders
304 Main Ave. S., Suite 300
Renton, WA 98055
(206) 271-4514
Visa, M/C welcome

Washington State residents add 6.6% tax.
Overseas shipping wt. 1 oz.
All foreign orders (incl. Canada) will require additional postage and shipping.

*Apple II+ is registered trademark of Apple Computer Inc.

REQUEST FOR CHANGE OF ADDRESS

Dear A.P.P.L.E.,
I am moving and my present address is:

PRESENT:

NAME _____
ADDRESS _____
CITY _____
STATE _____ ZIP CODE _____
EFFECTIVE DATE _____

NEW ADDRESS:

NAME _____
ADDRESS _____
CITY _____
STATE _____ ZIP CODE _____
MEMBERSHIP NUMBER _____

*Please fill out and return this notice two weeks before you move.

PRODUCTS NOW AVAILABLE TO A.P.P.L.E. MEMBERS

Call —A.P.P.L.E. Specials[5]

BIG MAC.LC[2]	$27.50
PLE Closeout	$20.00
Word Weaver ///[2]	$60.00

A.P.P.L.E. LIBRARY

AP. Drive Technical Manual	$30.00
"Apple Orchard" No. 1 (magazine)	$2.50
Apple Users Guide	$13.00
Beneath the Apple DOS	$17.00
"Best of Cider Press"	
(S.F. Newsletter Annual) Vol I	$4.50
(S.F. Newsletter Annual) Vol II	$9.50
Call —A.P.P.L.E. annuals	
1978	$7.50
1979	$10.00
1980	$15.00
1981	$18.50
Call —A.P.P.L.E. in Depth No. 1	$7.00
Docupak Vol 1 "The Wozpak II" (documentation)	$20.00
Inside Washington Apple Pi	$7.00
Power of VisiCalc	$8.50
Using 6502 Assembly Language	$17.00
What's Where in the Apple	$12.50
Wozpak diskette[1]	$17.50
Woz Talks (VHS Video cassette)	$27.50

AS IS SOFTWARE

Public domain diskettes (DOS 3.2.1) (Send 35¢ S.A.S.E. for list of titles)	ea. $4.00

Diskettes supplied on DOS 3.3 except as noted.

To avoid shipping delays, always order new products separately.

ORDER FORM ON REVERSE

A.P.P.L.E. DISKPAKS

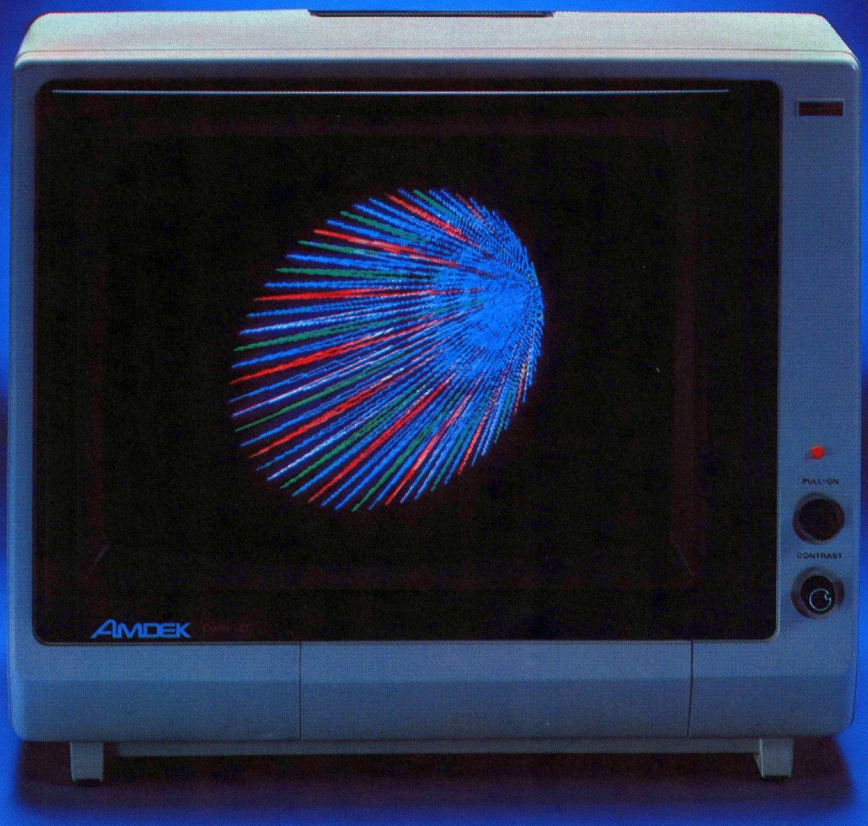
Call —A.P.P.L.E. Anthology Vols I-VI, incl.	ea. $12.50
Diskpak 12	$16.50
Diskpak 34A	$13.00
Diskpak 5	$13.00
Diskpak 6B (DOS 3.2 disk utility)	$18.50
Diskpak 6B-16 (DOS 3.3 disk utility)	$18.50
Diskpak 7	$17.50
Diskpak 8	$13.00
In Depth 1 Diskette	$16.50

Apple Prestige Product LinE

APfile — Text file utility	$18.50
ApController[4]	$95.00
AP.DRIVE	$395.00
Ap.DRIVE with Controller[4]	$475.00
Ap L Case 1	$38.50
Ap L Case 2 (Lower case chip)	$27.50
APmail II	$22.50
Applesoft Carpenter	$18.50
Aptest	$18.50
Aptest 3.3 Update	$5.00
Big Mac Macro-Assembler/Ted	$22.50
Big Mac.LC[6]	$28.50
Disk Banks	2 for $15.00
Higher Fonts 1	$13.00
Higher Text Plus	$27.50
Program Global Editor	$18.50
Program Line Editor	$25.00
Soft Seventy	$22.00
Symbol Symon	$20.00
WordWeaver ///	$75.00

[1]-$15.00 if ordered with Wozpak
[2]-Pre-release price
[4]-Available soon
[6]-Available late March
[7]-$7.50 with Ap L Case 1
$10.00 with Ap L Case 2

The High-Resolution Color Monitor for Apple II



Amdek's new Color-II monitor with DVM interface board

No video monitor has ever combined Apple II compatibility with exceptional performance like the new 13" Color-II monitor from Amdek.

Color-II features our optional new DVM peripheral board for easy interfacing with your Apple II. And look at the other top-of-the-line features Color-II offers:

- RGB, TTL input for high resolution graphics.
- 80 x 24 character display capability (with expansion card by Vydec).
- 560 (H) x 260 (V) resolution.
- Molded-in carrying handle for easier portability.
- Front-mounted controls for faster fine-tuning.

So if you want to get the most from your Apple II system, get the "most" monitor. Ask your dealer about Amdek's new Color-II monitor with the optional DVM interface board.

What is DVM? The DVM, or "Digital Video Multiplexor," is a low cost interface that allows the Apple II computer to be used with an RGB monitor, such as our Color-II.

Amdek's DVM is software programmable to allow transparent operation, and is parallel with existing Apple text and graphic modes. Three of its channels are used to multiplex the existing Apple text, low resolution and high resolution graphics. The 4th channel allows the use of an 80 character line video board.

Our DVM is also color channel software programmable, enabling you to turn the three color channels on or off by software control. For example, the red and blue channel can be turned off when the 80 character channel is turned on, resulting in a green phosphor video presentation.

For convenient operation, the DVM board may occupy any slot in the Apple II. The DVM also features low-power consumption and low-power schottky logic.

AMDEK

Amdek Corporation, 2420 E. Oakton Street, Suite "E" Arlington Heights, Illinois 60005 (312) 364-1180 TLX: 25-4786

THE NEXT GENERATION OF COMPUTER GAMES IS HERE

FROM THE MAKERS OF THE FIRST ANIMATED ADVENTURE GAMES

Synergistic Software

- **APVENTURE TO ATLANTIS** is a new fast action apventure game, by Robert Clardy, continuing the Odyssey Apventure series. The forces of sorcery are gathering to repulse the marauding scientific Atlantean Kingdom. The epic battle can end either in the enslavement of the world or the absolute destruction of Atlantis.
- Computer adventuring has been good, Synergistic Software has just made it better. Can you imagine the most exciting adventure game ever composed with more color and sound, highly detailed animation, arcade like action, true role playing, greater challenge, unending variety and endless enjoyment? That's what the Next Generation of computer games is all about. This professionally designed adventure was specifically written to take full advantage of all the Apple's capabilities. Requires careful planning and strategy, quick thinking and reflexes, as well as a 48K Apple II* with DOS 3.3 and paddles.

Only $40.00

*Apple II is a trademark of Apple Computer Inc.

From

Synergistic Software

5221 120TH AVE. S.E.
BELLEVUE, WA 98006
(206) 226-3216